AD-A048 769 COMPUTER SCIENCES CORP SAN DIEGO CALIF F/G 17/2

SIMULATION CODE FOR UNDERWATER TELEVISION SYSTEMS (SCOUTS).(U)

SEP 77 A GORDON, T SELDEN N00123-74-C-0312

UNCLASSIFIED NOSC-TR-146 NL

1 OF 1

AD

A048769

END

DATE

FILMED

2-78

DDC

AD A048769

12

NOSC

NOSC TR 146

NOSC TR 146

Technical Report 146

SIMULATION CODE FOR UNDERWATER TELEVISION SYSTEMS (SCOUTS)

A Gordon
Naval Ocean Systems Center
T Selden
Computer Sciences Corporation

1 September 1977

DDC
RECEIVED
JAN 19 1978
F

Interim Report: June 1976 — May 1977

Prepared For
Naval Sea Systems Command

Approved for public release; distribution unlimited.

DDC FILE COPY

NAVAL OCEAN SYSTEMS CENTER
SAN DIEGO, CALIFORNIA 92152

NAVAL OCEAN SYSTEMS CENTER, SAN DIEGO, CA 92152

AN ACTIVITY OF THE NAVAL MATERIAL COMMAND

RR GAVAZZI, CAPT USN
Commander

HL BLOOD
Technical Director

ADMINISTRATIVE INFORMATION

Work on this report was performed from 1 June 1976 to 1 May 1977. The work was funded as part of the Deep Ocean Technology Project by the Naval Sea Systems Command under Program Element 63713N. This report was reviewed for technical accuracy by Dr. Peter McCardell of the Naval Ocean Systems Center, San Diego. Work by CSC was performed under Contract N-00123-74-C-0312.

Released by
IP Lemaire, Head
Advanced Systems Division

Under authority of
HR Talkington, Head
Ocean Technology Department

(18) NOSC (19) TR-146

UNCLASSIFIED

SECURITY CLASSIFICATION OF THIS PAGE (When Data Entered)

REPORT DOCUMENTATION PAGE		READ INSTRUCTIONS BEFORE COMPLETING FORM
1. REPORT NUMBER NOSC Technical Report 146 (TR 146)	2. GOVT ACCESSION NO.	3. RECIPIENT'S CATALOG NUMBER
4. TITLE (and Subtitle) (6) SIMULATION CODE FOR UNDERWATER TELEVISION SYSTEMS (SCOUTS),		5. TYPE OF REPORT & PERIOD COVERED (9) Research and Development rept. 1 Jun[redacted]76[redacted]1 May [redacted]77,
		6. PERFORMING ORG. REPORT NUMBER
7. AUTHOR(s) (10) Alan Gordon Naval Ocean Systems Center Thomas Selden Computer Sciences Corporation		8. CONTRACT OR GRANT NUMBER(s) (15) N00123-74-C-0312 New
9. PERFORMING ORGANIZATION NAME AND ADDRESS Naval Ocean Systems Center San Diego, CA 92152		10. PROGRAM ELEMENT, PROJECT, TASK AREA & WORK UNIT NUMBERS Program Element 63713N
11. CONTROLLING OFFICE NAME AND ADDRESS Naval Sea Systems Command Washington, D.C.		12. REPORT DATE (11) Sept[redacted]77
		13. NUMBER OF PAGES (12) 44p.
14. MONITORING AGENCY NAME & ADDRESS (if different from Controlling Office)		15. SECURITY CLASS. (of this report) UNCLASSIFIED
		15a. DECLASSIFICATION DOWNGRADING SCHEDULE

16. DISTRIBUTION STATEMENT (of this Report)

Approved for public release; distribution unlimited.

DDC
RECEIVED
JAN 19 1978
F

17. DISTRIBUTION STATEMENT (of the abstract entered in Block 20, if different from Report)

18. SUPPLEMENTARY NOTES

19. KEY WORDS (Continue on reverse side if necessary and identify by block number)

underwater television
computerized simulation
FORTRAN IV

20. ABSTRACT (Continue on reverse side if necessary and identify by block number)

Based on the system performance equations developed in the Naval Undersea Center's NUC TP 303, Handbook of Underwater Imaging System Design, a computer code was written to predict the performance of underwater television systems. This code, known as SCOUTS (for "Simulation Code for Underwater Television Systems") is written in FORTRAN IV and is suitable for demand or batch runs. SCOUTS allows the user to select environmental, geometric and hardware parameters. It can be used for system design as well →

DD FORM 1473 1 JAN 73 EDITION OF 1 NOV 65 IS OBSOLETE

UNCLASSIFIED

SECURITY CLASSIFICATION OF THIS PAGE (When Data Entered)

390612

UNCLASSIFIED

SECURITY CLASSIFICATION OF THIS PAGE(When Data Entered)

20. Continued.

as analysis. SCOUTS includes a parametric mode which allows automatic variation of multiple parameters. A complete source listing is included as an Appendix.

UNCLASSIFIED

SECURITY CLASSIFICATION OF THIS PAGE(When Data Entered)

SUMMARY

PROBLEM

Using the system performance equations developed in NUC TP 303, Handbook of Underwater Imaging System Design, develop a computer code for predicting the performance of underwater television systems.

RESULTS

A code called SCOUTS (for "Simulation Code for Underwater Television Systems") and suitable for batch or demand runs has been written in FORTRAN IV. SCOUTS allows the user to select environmental, geometric, and hardware parameters. It can be used for system design as well as analysis. SCOUTS includes a parametric mode that allows automatic variation of multiple parameters.

BY
DISTRIBUTION/AVAILABILITY CODES
SPECIAL
A

CONTENTS

LIST OF FIGURES

LIST OF TABLES

1.0 INTRODUCTION

This report documents the Naval Ocean Systems Center's (NOSC) Simulation Code for Underwater Television Systems (SCOUTS). SCOUTS is a real time, interactive (when used in the DEMAND mode) computer code which is useful in the design, analysis and performance prediction of a large class of underwater television systems. A complete printout of SCOUTS in the FORTRAN IV language is included in the Appendix. For those with immediate applications, the information in Sections 2 and 5 is sufficient to describe the operation and use of SCOUTS. Sections 3 and 4 give the theoretical basis for the calculations and internal program structure, respectively. The remainder of this Section concerns itself with the background and rationale of SCOUTS' development.

1.1 NEED FOR PERFORMANCE MODELING OF UNDERWATER TELEVISION

During the past decade and a half man has rapidly extended his presence beneath the surface of the ocean. Manned submersibles, tethered vehicles, moored drill rigs and even underwater habitats reflect his desire to understand and utilize the unique world beneath the ocean's surface. Underwater photography has been used throughout this era to document subsurface missions. This time period has also seen the development of underwater television to the point that it is equal in importance to photography as an undersea visual sensor.

Modern underwater television systems use lenses, light sources and cameras specifically designed for this environment. These components are incorporated into system concepts which attempt to minimize backscatter while maintaining an acceptable signal-to-noise ratio. As a result of the progress made during the last 15 years, the designer of underwater television systems now can choose from a wide variety of components and techniques in tailoring the system to his particular requirements.

One rather disturbing effect of the availability of advanced components and techniques was that predicting the performance of the resulting system became much more difficult. Simple single scattering models gave erroneous results at the longer ranges corresponding to advanced imaging systems. More exact models were available, but they were primarily theoretical and not generally suitable for engineering use. Thus, by the early 1970s, a situation arose where hardware was available which appeared to promise improved system performance, but the methodology for comparing the available hardware and accurately predicting performance was lacking.

1.2 THE HANDBOOK OF UNDERWATER IMAGING SYSTEM DESIGN

Under the sponsorship of the Navy's Deep Ocean Technology Program, the Naval Ocean Systems Center undertook the task of providing a definitive handbook for the designer of underwater television systems. This effort produced the Handbook of Underwater Imaging System Design (HUISD) (Reference 1). HUISD provided a detailed discussion of optical water parameters, propagation of light underwater, characteristics of system components and

PRECEDING PAGE BLANK-NOT FILMED

advantages and drawbacks of system concepts then in use or proposed. More importantly, HUISD gave a detailed design procedure for obtaining measures of system performance based on the above factors. This design procedure, validated by comparative system tests performed by NOSC at Morris Dam in 1970, was presented both as a set of nomograms and a step-by-step series of equations. These equations, which were more accurate and flexible than the nomograms, required lengthy hand calculations and were tedious to implement. Consequently, these system equations were not applied as widely as had been hoped, though their predictions proved accurate in those instances where the hand calculations were performed.

1.3 EVOLUTION OF SCOUTS

About a year after the publication of HUISD, NOSC began work on the computer modeling portions of the Advanced Unmanned Search System (AUSS) program. Part of the AUSS model required the simulation (i.e., performance prediction) of underwater television systems. Since the HUISD equations were already developed and validated, these equations were used as the basis for the AUSS computer code. Only the HUISD performance analysis of conventional systems (i.e., systems using non-laser sources and raster-scanned receivers) was coded, since these systems were the ones most commonly in use. The first AUSS TV program was coded in BASIC, but in order to achieve faster operating times the program was rewritten in the FORTRAN language.

Experience with the AUSS TV program quickly showed that the HUISD system performance equations, coupled with real time, interactive demand terminal computing, provided a powerful tool for TV system design and analysis. Using the program the operator was immediately furnished with the performance of the selected configuration as well as indications as to what (e.g., source power, backscatter) were the limiting factors. By successively adjusting the TV system's configuration the operator could optimize it for his particular arrangement. This procedure was successfully used in designing and validating systems used in search, small-object detection, obstacle avoidance, etc.

One of the problems with the AUSS TV program was that it is a subprogram integral to the much larger AUSS model. In order to use just the TV portion, much information extraneous to the operation of the TV subsystem has to be keyed in. Because there was a significant number of AUSS users who were interested primarily in the TV subprogram, it was decided to develop SCOUTS as an independent computer code for the simulation of underwater television.

SCOUTS differs in a few important ways from the AUSS TV subprogram. SCOUTS is coded in FORTRAN IV and should be compatible as written with most medium- and large-size computers. A parametric capability allows several variables to be automatically varied according to limits defined by the user. Finally, although SCOUTS is primarily demand terminal-oriented, a batch option is included for computer facilities lacking real time capabilities.

2.0 OVERVIEW OF SCOUTS

This section will be concerned with defining the problem which SCOUTS solves, the inputs required and the outputs provided. Coupled with the detailed operating instructions in Section 5, this section provides sufficient information for successful operation of the program.

2.1 SYSTEM GEOMETRY

The geometry assumed by SCOUTS for its calculations is shown in Figure 1. The source and receiver are spatially separated by the source-receiver separation, d. Without loss of generality the source is assumed to be to the receiver's right when looking out towards the target. Their optical axes are coplanar and each is "canted in" by an angle δ so that a perpendicular dropped from the axes' intersection point will bisect the line joining the source and receiver. The receiver's (i.e., TV's) horizontal scan direction is parallel to the line joining source and receiver.

The target's center is in the plane containing the source and receiver's optical axes (i.e., the plane of the paper). The long dimension of the object is in the plane which is orthogonal to the plane of the paper and parallel to the line joining source and receiver. The

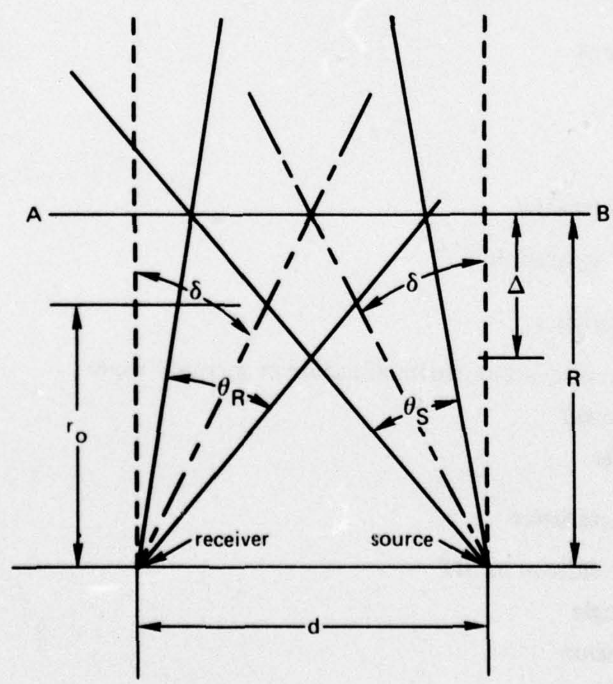

Figure 1. SCOUTS system geometry.

point of the object most distant from the source-receiver plane (i.e. the plane containing the source and receiver and perpendicular to the plane containing their optical axes) is assumed to be situated a distance R from it. The height of the object, H, is the extent of the object perpendicular to the source-receiver plane.

The distance r_0 in Figure 1 is the distance past which backscatter contributes to the image. The depth of field, Δ, is the maximum distance from the target plane at which there is a "common volume" where source and receiver beams intersect any portion of the target whose height exceeds Δ will not be imaged.

2.2 USER INPUTS

Table 1 contains the inputs necessary to define the imaging problem for SCOUTS. All these variables must be entered on the initial calculation of any run. A rewrite capability (see Section 2.5) is included so that this information does not have to be entered each time it is desired to change only some of the variables. An example of the initial input for a demand run is shown in Figure 2.

Table 1. User inputs.

	Environment
1.	Water type: coastal or deep
2.	Peak-to-trough wave height
3.	Peak-to-trough bottom roughness
	Target Dimensions
4.	Target length
5.	Target height
	Platform Characteristics
6.	Source-receiver separation
	Source Characteristics
7.	Source type: incandescent, thallium iodide or mercury vapor
8.	Source input power
9.	Source full angle
	Receiver Characteristics
10.	Receiver type: vidicon or SIT
11.	Receiver full angle
12.	Optics transmission
13.	F-number

```
                    CONVENTIONAL TV SYSTEM
   ENVIRONMENT
 COASTAL OR DEEP, 1 OR 2:?
>1
 WAVE HEIGHT-PEAK TO TROUGH,FT:?
>2.
 BOTTOM ROUGHNESS-PEAK TO TROUGH,FT:?
>.5
   TARGET DIMENSIONS
 LENGTH OF TARGET,FT:?
>10.
 HEIGHT OF TARGET,FT:?
>.5
   PLATFORM CHARACTERISTICS
 SOURCE RECEIVER SEPARATION,FT:?
>1.
   SOURCE CHARACTERISTICS
 INCANDESCENT, THALLIUM IODIDE, OR MERCURY VAPOR; 1, 2 OR 3:?
>2
 SOURCE POWER,WATTS:?
>200.
 FULL SOURCE ANGLE,DEG:?
>40.
   RECEIVER CHARACTERISTICS
 VIDICON OR SIT, 1 OR 2:?
>1
 FULL RECEIVER ANGLE,DEG:?
>40.
 TRANSMISSION OF OPTICS:?
>.9
 F-NUMBER:?
>1.5
```

Figure 2. Sample input for demand run.

Table 2 is for the most part self-explanatory; however, a few comments are in order. SCOUTS was originally written for an object lying on the bottom viewed from a TV camera which was responding to ocean heave. To ensure an adequate depth of field, Δ, SCOUTS chooses Δ so that

$$\Delta = H + 0.5 * (h_1 + h_2) \tag{1}$$

where h_1 and h_2 are the wave height and bottom roughness respectively. For situations where the target is not on the bottom, h_1 and h_2 should be set to zero.

SCOUTS incorporates information from HUISD to provide the appropriate characteristics for the water, lamp and receiver types. These characteristics will be discussed in detail in Section 3. However, in order for the user to supply meaningful inputs, it is necessary to discuss some of these characteristics here.

The two water types, coastal and deep, have optical characteristics identical with those plotted in Figures 6.2(B) and 6.2(C) of HUISD. The coastal water has a minimum attenuation, α, of 0.253/m ($1/\alpha$ = 3.95 m) at λ = 540 nm. For the deep water, the corresponding numbers are α = 0.049/m ($1/\alpha$ = 20.4 m) at 475 nm. The scattering coefficient, s, is 0.238/m for the coastal water and 0.030/m for deep water.

For sources with conical beam patterns, the required "full source angle," θ_S, is just the apex angle of the in-water beam pattern. For a nonconical source, the "equivalent conical beam pattern" (HUISD, p. 6-11, Eq. 6.10) must be determined and its apex angle used as *input*. For "source power," SCOUTS requires the electrical power input to the lamp, then computes the light power output.

The full receiver angle, θ_R, is the full in-water horizontal field of view of the TV camera. SCOUTS assumes the standard 4-by-3 aspect ratio for the receiver's field of view.

For transmission of optics SCOUTS requires the overall decimal efficiency of the source and receiver. This is the combined efficiency of all the optical elements following the source and up to and including the camera lens. For example, if the collection efficiency of the light source and the transmissivity of the camera lens were each 0.9, the required optical transmissivity would be 0.81.

The f-number, f/, is defined for small angles by

$$f/ = \frac{f}{D} \qquad (2)$$

where f is the focal length and D the receiver aperture diameter. The f/ has a minimum value of 0.5. Most camera lenses will indicate the f/ corresponding to each aperture stop.

2.3 TYPES OF CALCULATIONS AVAILABLE

2.3.1 Demand and Batch Runs

SCOUTS can be run either as a batch or demand run. It is important to remember that SCOUTS was designed for real time, interactive computing, i.e. for demand runs. Design problems are most efficiently handled in the demand mode, since the user will be able to refine his design based on immediate feedback of the performance of his previous choice. Another advantage is that after a period of running design problems via demand runs, the user will develop his intuition for good underwater systems design.

The batch run capability was included primarily for computer systems not having real time facilities. However, the batch mode can be used to advantage in some situations. Some computer facilities charge substantially less for batch runs than for real time operation. Although SCOUTS' costs are low because less than 1.5 sec combined CAU (central arithmetic unit) and ER (executive request) time per calculation is required on the NOSC Univac 1110, the cost savings might be significant if many calculations are required. Batch runs necessarily provide a hard-copy output; this output may be absent or of poor quality on demand terminals. In very large runs using the parametric mode (see Section 2.3.2) where many variables are to be parametized in a noninteractive fashion, using the batch mode allows the program to run without the operator being present.

2.3.2 Nonparametric and Parametric Modes

Nonparametric and parametric modes are available for both demand and batch runs. In the nonparametric mode, each input variable (see Table 1) takes on only one value and a calculation is immediately performed. In the parametric mode, the user selects one or more variables which he parametizes by assigning each a beginning value, incremental step and final value. SCOUTS then performs as many calculations as necessary to evaluate the system for all the selected values of the input variables, and the results are output.

The parametric mode is thus similar to a succession of nonparametric calculations. The user selects the mode appropriate for his objectives. In optimizing a system, a succession of nonparametric calculations will allow the user to select new input values based on the results of past calculations. The performance of a selected system for various values of input data can be documented more quickly in the parametric mode.

2.3.3 Calculational Modes

There are three calculational modes available to the user: single range, maximum range or maximum swath width. The calculational mode is chosen in demand runs by the appropriate response to a SCOUTS question and in batch runs by inclusion of the appropriate mnemonic (see Table 4). In the single-range mode the system performance is evaluated at the range supplied by the user. To understand the other two modes we must first define what is meant by usable swath width.

The usable (or actual) swath width at range R is measured along that line in the source-receiver-target plane which is parallel to the line joining source and receiver and a distance R from it (the line AB in Figure 1). A segment of this line will have the following two properties: (1) a target located on this segment will have an image contrast greater than or equal to 0.07 and (2) the target's image will be large enough to span eight or more resolved TV lines. The length of this segment is taken to be the actual swath width. The actual swath width thus represents the lateral dimension in which successful imagery is possible.

As a function of range the swath width increases almost linearly with range, reaches a maximum and then drops off rapidly as the range is increased further. The maximum range calculation evaluates the system at that range where the swath width, after having passed its maximum, drops to zero. Similarly, the maximum swath width calculation evaluates the system at the range which yields the greatest swath width.

In the nonparametric mode, SCOUTS can evaluate the system in any of the three calculational modes. In the parametric mode, the user can have the system evaluated at either the maximum range or maximum swath width by indicating his choice and setting the range equal to zero. If the range has some other value the system will be evaluated at the selected range; however, SCOUTS will, at the user's option, calculate the value of the maximum range, maximum swath width, or both. Since SCOUTS performs these maximizing calculations by performing a series of single-range calculations, running time and costs can be minimized by selecting maximizing calculations only when necessary.

2.4 OUTPUT

2.4.1 Minimum Range

If the target is located too close to the source-receiver plane TV performance can be degraded. At distances which are too close, the resultant depth of field becomes less than the minimum value required by Eq. 1. Additionally, at distances sufficiently short the source will be pointing directly towards the receiver and blind it. SCOUTS therefore calculates a minimum range prior to evaluating the user's system. In demand runs this information is presented to the user in time for the user to choose a range greater than the minimum. SCOUTS then checks that the input range is indeed greater than the minimum. If not, the calculation will not be performed.

One of the disadvantages of batch runs is that the user does not receive an indication of the allowable minimum range until the run is over. Thus, there is an uncertainty as to the smallest range that will execute sucessfully. Since SCOUTS sets the minimum range no smaller than the source-receiver separation, batch input ranges should be larger than this value. This will not guarantee a successful run but is the best the user can do with no other *a priori* information.

2.4.2 Summary Output

The results of a SCOUTS calculation (or calculations in the parametric mode) represent the performance evaluation of the specified TV system. This performance evaluation is provided by SCOUTS in the form of two output summaries, the intermediate summary (Figure 3) and the final summary (Figure 4). Intermediate summaries are provided at the end of each demand nonparametric run, while final summaries are output in all modes. Both summaries contain essentially the same information in different formats. The meaning of each of the outputs will now be presented in the order in which they appear on the final summary.

The first thirteen columnar entires on Figure 3, from WATER TYPE to F-NUMBER, reproduce the user-selected input so that a record of the evaluated system is available along with the output. RANGE, FT and RANGE, AL give the range at which the calculation was performed in feet and attenuation lengths, respectively. All the performance measures from AVAILABLE LINES AT CTR to the second to the last entry refer to these measures evaluated at this range. MINIMUM, FT is the minimum allowable range (See Section 2.4.1) in feet. If the maximum range or maximum swath width options are chosen, the range at which each of these occurs is printed out in feet after MAXIMUM, FT and BEST, FT, respectively.

The number of resolved TV lines spanning the object length is given by AVAILABLE LINES AT CTR for the object located at the center of the receiver's field of view and by AT EDGE for the object located at the right-hand edge of the receiver's field of view. Because of the interpolation subroutine used, the number of lines may be in excess of 525, which is taken to be the display limit. It has been shown that eight lines represent the threshold of object recognition (Reference 3).

SCOUTS calculates the image contrast for an object having an inherent contrast of 50 percent and a highlight reflectivity of 75 percent. CONTRAST AT CENTER and AT EDGE are respectively the image contrast for a target located at the center and right-hand edge of

```
                        CONVENTIONAL TV SYSTEM
WATER TYPE:            COASTAL              SOURCE:              THAL. IOD.
WAVE HEIGHT,FT:    2.00000                  SOURCE POWER:      200.00000
BOTTOM,FT:          .50000                  SOURCE BEAM:        40.00000
LENGTH,FT:        10.00000                  RECEIVER:            VIDICON
HEIGHT,FT:          .50000                  RECEIVER BEAM:      40.00000
TAU:                .90000
S.R. SEP.:         1.00000                  F/:                  1.50000

RANGE,FT:         24.40228                  RANGE,AL:            1.87424

AVAILABLE LINES AT CTR.: 393.90030          AT EDGE:           393.90030
CONTRAST AT CENTER:           .16397        AT EDGE:              .07523

AVAILABLE L H S SW WIDTH = 8.81963 FT        ACTUAL = 8.22299 FT
LIMITED BY:R. ANG

AVAILABLE R H S SW WIDTH = 8.95223 FT        ACTUAL = 8.22299 FT
LIMITED BY:S. ANG

AVAILABLE SW WIDTH = 17.77186 FT           ACTUAL = 16.44598 FT
```

Figure 3. Intermediate summary.

the receiver's field of view. Since backscatter is most severe towards the right edge of the field of view (remember that SCOUTS assumes the light is to the right of the receiver) the contrast at this point (as well as the number of lines available) will always be less than that in the center.

The next eight entries on the final summary give information on the swath width to be expected. AV L.H.S. (R.H.S.) SW WIDTH, FT is the available left-hand (right-hand) swath width. This is the segment, in feet, of the swath width line (See Section 2.3.3) to the left (right) of the center of the receiver's field of view which is illuminated by both source and receiver beams. ACTUAL, FT is the portion of the available left-hand (right-hand) swath width, in feet, where the target may be successfully imaged. Again, the three criteria used by SCOUTS for successful imaging are: (1) adequate depth of field as defined by Eq. (1), (2) at least eight resolved TV lines across the target's longest dimension and (3) an image contrast of at least 7 percent.

LIMITED BY gives a mnemonic which indicates what is limiting the extent of the R.H.S. (L.H.S.) actual swath width. The mnemonics S. ANG. and R. ANG refer to source angle and receiver angle respectively and indicate that the corresponding beam's geometry is limiting the available swath width. BKSKTR indicates that the available swath exceeding 7-percent contrast is the limiting factor. Similarly, PWR signifies that the available swath width was limited by the requirement for eight TV lines/object length. DISPLY is printed when the number of lines/raster height required to place eight TV lines across the image's longest dimension exceeds 525.

AV SWATH WIDTH, FT is the arithmetic sum of the available left- and right-hand swath width and indicates the total available swath width for the given geometry and range. ACTUAL, FT is the corresponding sum of the actual swath widths, which is the linear

WATER TYPE:	COASTAL	COASTAL	COASTAL	COASTAL
WAVE HEIGHT,FT:	2.0000	2.0000	2.0000	2.0000
BOTTOM ROUGHNESS,FT:	.5000	.5000	.5000	.5000
TARGET LENGTH,FT:	10.0000	10.0000	10.0000	10.0000
TARGET HEIGHT,FT:	.5000	.5000	.5000	.5000
S.R.SEP.,FT:	1.0000	1.0000	2.0000	3.0000
SOURCE:	THAL. IOD.	THAL. IOD.	THAL. IOD.	THAL. IOD.
SOURCE POWER,WATTS:	200.0000	200.0000	200.0000	200.0000
SOURCE BEAM,DEG:	40.0000	40.0000	40.0000	40.0000
RECEIVER:	VIDICON	VIDICON	VIDICON	VIDICON
RECEIVER BEAM,DEG:	40.0000	40.0000	40.0000	40.0000
TAU:	.9000	.9000	.9000	.9000
F-NUMBER:	1.5000	1.5000	1.5000	1.5000
RANGE,FT:	24.4023	24.4023	29.7573	33.2921
RANGE,AL:	1.8742	1.8742	2.2855	2.5570
MINIMUM,FT:	2.5837	2.5837	3.0139	3.2687
MAXIMUM,FT:	.0000	.0000	.0000	.0000
BEST,FT:	24.4023	24.4023	29.7573	33.2921
AV LINES AT CTR:	393.9003	393.9003	322.7560	288.1938
AT EDGE:	393.9003	393.9003	322.7560	271.0501
CONTRAST AT CTR:	.1640	.1640	.1536	.1469
AT EDGE:	.0752	.0752	.0659	.0601
AV L.H.S. SW WIDTH,FT:	8.8196	8.8196	10.7119	11.9459
ACTUAL,FT:	8.2230	8.2230	10.1408	11.3969
- LIMITED BY:	R. ANG	R. ANG	R. ANG	R. ANG
AV R.H.S. SW WIDTH,FT:	8.9522	8.9522	10.9773	12.3444
ACTUAL,FT:	8.2230	8.2230	10.1408	10.2479
- LIMITED BY:	S. ANG	S. ANG	S. ANG	BKSKTR
AV SWATH WIDTH,FT:	17.7719	17.7719	21.6893	24.2904
ACTUAL,FT:	16.4460	16.4460	20.2816	21.6448
BEST,FT:	16,4460	16.4460	20.2816	21.6448

Figure 4. Final summary.

measure of the total horizontal distance over which satisfactory imagery is possible. BEST, FT is the maximum actual swath width. Zero will appear for this quantity unless the maximum swath width option has been selected.

3.0 BASIS OF SCOUTS CALCULATIONS

This section will outline the main features of the SCOUTS performance analysis and relate them to the corresponding steps in HUISID. This information is not strictly necessary for the general SCOUTS user, but it is useful to those who might desire to alter some of the program's internal parameters to suit their specific problems. To that end, the various water and hardware characteristics used by SCOUTS will be reviewed in Sections 3.1 and 3.2. Section 3.3 will provide an outline of the calculational flow keyed to the labeled line numbers of the detailed listing in the Appendix.

3.1 WATER CHARACTERISTICS

All the water characteristics used by SCOUTS are documented and referenced in HUISID and appear as stored matrices entered via DATA statements in the program MAIN. The following discussion of the program's features requires reference to the appendix, where the subroutines are arranged in alphabetical order. MAIN is the main program; the rest are subroutines. The function of each is explained in Section 4.

The array A(I, J) (line MAIN 21) stores the attenuation coefficient for coastal (I=1) and deep (I=2) water for forward (J=1) and backscattered (J=2) light. It was obtained from HUISD (Table 6.12). The array B(I,J,K) (MAIN 24) contains the ratio of the effective to actual attenuation coefficient indexed according to water type (I), forward or reverse propagation (J) and full beam angle (K); these data also come from Table 6.12 in HUISD. The interpolation array of corresponding beam angles is stored in matrix T (MAIN 49). Scattering coefficients, which were also obtained from Table 6.12, are stored in matrix S (MAIN 46). For both coastal and deep waters a rear hemisphere scattering (i.e., backscattering) percentage of 2 percent is used (line TVFU 33).

3.2 HARDWARE CHARACTERISTICS

The spectral characteristics of the source, receiver and water enter into the HUISD systems performance analysis through the constructed functions $G(\lambda)$ and $H(\lambda)$, which refer to forward and backscattered light, respectively. These functions are stored in the SCOUTS arrays G(I,J,K) and H(I,J,K) where I, J, K refer to water, receiver and source types, respectively. These entires were computed by subtracting entries in Table 6.8 of HUISD in accordance with the effective bandwidths listed in Table 6.5.

The imaging capability of the two types of cameras are stored in the array FM (I, J) (MAIN 30), where I refers to receiver type (i.e. I = 1 = vidicon and I = 2 = SIT) and J indexes two constants per receiver type. For a given camera tube current, i, the number of resolved TV lines, N, is obtained according to

$$N = \frac{\ln\left(i/FM(I,1)\right)}{FM(I,2)}\,. \tag{3}$$

The values of FM(I, J) were obtained by fitting Eq. (3) to available camera tube data. For the SIT the data were taken from HUISD, Figure 6.4(E), and the vidicon data from Reference 2. In each case the 50-percent contrast curve was used.

3.3 PERFORMANCE CALCULATIONS

The first calculation SCOUTS performs for each system's performance analysis is the evaluation of the minimum range. This occurs in lines HLIM 25-47. HLIMIT sets the minimum range equal to the greatest of (1) the source-receiver separation, (2) the minimum range which gives the required depth of field and (3) the minimum range which precludes the source and receiver looking at each other.

All the other calculations are done in the subroutine TVFUNC and the routines called by it. After some initialization, TVFUNC sets the range equal to that desired for single-range calculation (line TVFU 53) or to 110 percent of the minimum range in the maximizing modes (line TVFU 55). Next, a check is performed to make sure the display limit is satisfied (TVFU 59-64). If not, the remaining calculations are skipped and a flag is set so that a warning that the display limit has been exceeded is printed in the summary.

In lines TVFU 72-120, SCOUTS evaluates the number of lines at the center and the edge of the field of view. If less than the minimum (i.e. eight) number of lines is obtained at the center of the field of view, SCOUTS assigns minimum left- and right-power-limited swath widths (TVFU 101-105). If there is no power limit even at the edge of the field of view, the maximum swath widths are associated with the power limit (TVFU 117-119). In the case where the eight-line limit is encountered at some angle other than the maximum or minimum, this angle is found in an iterative search (TVFU 87-99) and the appropriate swath widths are computed (TVFU 109-114). In every case, the photocathode current is obtained in subroutine DFUNC, which implements Eqs. 6.17 and 6.34 of HUISD. EFUNC evaluates the number of resolved TV lines, using the matrix FM(I, J).

The effects of backscatter are accounted for in lines TVFU 121-127. Using the subroutine XFUNC, which evaluates the photocathode current due to backscattered light (Eq. 6.48 of HUISD), the contrast is calculated at the center and at the right-hand edge of the field of view. Using the interpolation routine, GFUNC, an appropriate right-hand swath width due to backscatter, W(7), is obtained. The geometric limitations on the swath width are next obtained (W(2), W(3), W(5), W(6)) through the use of UFUNC and VFUNC. FNK finds the limiting right and left swath widths and descriptors. The limiting swath widths are summed to yield the total actual swath width.

Lines TVFU 146-184 determine whether the calculation is for a single-range or a maximizing calculation. If it is for a single-range calculation, TVFUNC returns control to HLIMIT. If not, TVFUNC determines whether the calculation is complete. If the last range increment is less than 5 percent of an attenuation length, control is returned to HLIMIT. If not, the range is incremented or decremented depending on the previous value of the swath width, and the performance is recalculated starting at line TVFU 40.

4.0 PROGRAM STRUCTURE

4.1 SUBROUTINE ORGANIZATION

SCOUTS consists of a main program and seventeen subroutines. The main program defines constants used in performing the calculations and acts as a program driver. Eleven of the seventeen subroutines perform the computations. The remaining six subroutines – SUBROUTINE DATAIN, SUBROUTINE CONVT, SUBROUTINE HLIMIT, SUBROUTINE UPDATE, SUBROUTINE DATAOT, and SUBROUTINE SUMMARY – were developed to provide flexibility of input and output. Figure 5 illustrates program hierarchy. A printout of each subroutine and of the main program is provided in the appendix.

SUBROUTINE DATAIN reads user-supplied input data and sets program control flags. If batch processing is desired, the user-supplied data are input via the card reader (unit number 5). Beginning in column one, the first card of this data deck must contain the five-character alphanumeric word BATCH. This card sets the flag for batch processing. Subsequent data cards are then read in an A6, 3X, 3F10.0 format. The alphanumeric information is a mnemonic instruction that directs the program to a specific section of code. The data values, where applicable, represent the initial, incremental, and final values of the data variables. Default values are set prior to reading any user-supplied data.

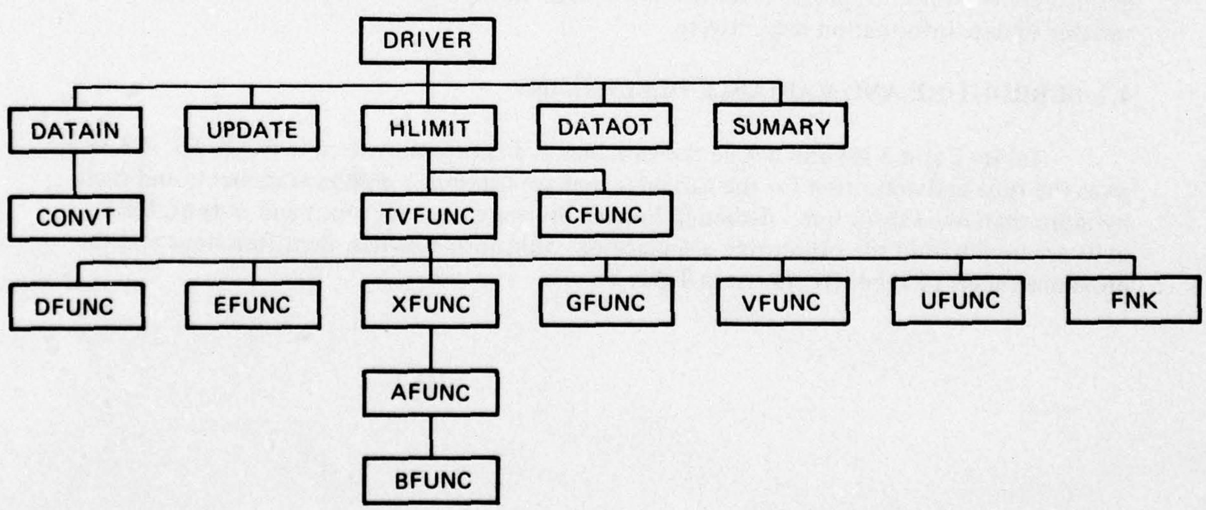

Figure 5. Program hierarchy.

If the program is being executed via a demand terminal, the terminal itself is the input device. In this case, interactive communication with the program is performed. The format of the input data, unless otherwise specified by the program, is G10.0. It should be noted that the program can be executed via a demand terminal without interactive communication by typing the word BATCH as the initial response. Data are input as though batch processing were being performed. (For more detailed information on input data, see the section under program operation.)

SUBROUTINE HLIMIT computes the minimum range and establishes the type of computations that will be made – single range, maximum range and/or maximum width.

Incrementing of the input data variables is performed by SUBROUTINE UPDATE. This subroutine is called only when parametric variables have been defined.

Two subroutines handle the printing of output data. Whenever execution is performed via a demand terminal, and no parametric variables have been defined, SUBROUTINE DATAOT prints the results after each evaluation is performed. This subroutine also writes a summary of the performance evaluation to the temporary disk file (unit number 9). At the conclusion of execution, SUBROUTINE SUMMARY reads the information stored on disk and prints tabular summaries of all performance evaluations computed. For batch processing, the printer (unit number 6) is the peripheral device utilized for output; for demand terminal processing, the terminal is the output device.

SUBROUTINE CONVT was developed to allow the demand terminal user to input integer values left-justified. If a two-digit integer value has been entered, the second digit is read as alphnumeric data and "converted" to the appropriate numeric value.

Data communication within the program is handled primarily through labeled common blocks. The four blocks established – OPTS, TVCOM, IOLIST, and UPD – store user-defined control options, program computing constants, input/output variables, and input variable update information respectively.

4.2 SUBROUTINE AND VARIABLE DEFINITIONS

Tables 2 and 3 list and define the variables and subroutines used in SCOUTS. Table 2 gives the type and definition for the variables that are listed in common statements and used by more than one subroutine. Although English units are used for input and output, MKS units are used within the subroutine calculations. Subroutine names, their functions and the programs called by them are shown in Table 3.

Table 2. Variable names and definitions.

Descriptive Variable	Type	Definition
NVIRON	integer	water type
WAVEHT	real	wave height
BTMRUF	real	buttom roughness
TARGLN	real	target length
TARGHT	real	target height
HEIGHT	real	height (or range)
SRSEP	real	source-receiver separation
SPOWR	real	source power
SDELF	real	full source angle
IRTPE	integer	receiver type
RDELF	real	full receiver angle
OPTRAN	real	optics transmission
FNUM	real	f-number
ISTPE	integer	source type
RANGMN	real	minimum range
RANGMX	real	maximum range
RANGEB	real	best range
SWWTHB	real	best swath width
HTUS	real	usual height
SWWTH	real	actual swath width
SWWTHL	real	actual L.H.S. swath width
SWWTHR	real	actual R.H.S. swath width
LIMFLG	integer	display limit flag
NDEXWL	integer	L.H.S. width index
NDEXWR	integer	R.H.S. width index
LINCEN	real	available lines at center
LINEDG	real	available lines at edge
CTRCEN	real	contrast at center
CTREDG	real	contrast at edge

Table 3. SCOUTS subroutines.

Subprogram	Function	Reference	Referenced by
MAIN	program driver; defines computing constants	DATAIN, HLIMIT, DATAOT, UPDATE, SUMARY	–
DATAIN	reads input data; sets control flags	CONVT	MAIN
HLIMIT	computes minimum range, defines type of computation	CFUNC, TVFUNC ATAN, TAN	MAIN
DATAOT	prints results after each evaluation; writes summary data to disk	–	MAIN
UPDATE	increments parametric input variables	–	MAIN
SUMARY	prints tabular summaries	–	MAIN
TVFUNC	main computational subroutines	DFUNC, EFUNC, VFUNC, XFUNC, GFUNC, UFUNC, FNK COS, ATAN, TAN	HLIMIT
CONVT	converts a single character to numeric data	–	DATAIN
XFUNC	calculates backscatter current	AFUNC	TVFUNC
AFUNC	Simpson's rule integration routine for backscatter integral	BFUNC, EXP, ALOG	XFUNC
BFUNC	integrand for backscatter integral	EXP	AFUNC
CFUNC	finds effective attenuation coefficient by interpolation	–	HLIMIT
DFUNC	computes signal current	TAN, ATAN, COS, EXP	TVFUNC
EFUNC	evaluates number of lines	ALOG	TVFUNC
FNK	interpolates for limiting width and description	–	TVFUNC
GFUNC	interpolates for backscatter width	TAN, ALOG	TVFUNC
UFUNC	calculates l.h.s. receiver and r.h.s. source widths	–	TVFUNC
VFUNC	calculates r.h.s. receiver and l.h.s source widths	–	TVFUNC

5.0 PROGRAM OPERATION

This section is intended to provide the user with information that is required for successful execution of the visual search system program.

5.1 PRELIMINARY REQUIREMENTS

The visual search system program utilizes three peripheral devices for transmission of data: the card reader, the printer, and a disk. The card reader and the printer are assigned the standard FORTRAN unit numbers 5 and 6 respectively. Unit 9 is assigned to a disk file used temporarily to store unformated data of summary output information. As required by the individual operating system involved, these unit numbers must be assigned to the appropriate peripheral device prior to compilation of the program.

After having assigned unit numbers to the appropriate peripheral device, the program is compiled and linked to create a set of absolute (or executable) binaries. Figure 6 illustrates the procedure used on the UNIVAC 1110 to create the absolute element for execution. In the example the absolutes are stored under the name VSEARCH of the file TVSENS. The program driver and each subprogram are stored under separate file element names.

```
@FOR,N  TVSENS.MAIN
@FOR,N  TVSENS.DATAIN
@FOR,N  TVSENS.CONVT
@FOR,N  TVSENS.HLIMIT
@FOR,N  TVSENS.TVFUNC
@FOR,N  TVSENS.UPDATE
@FOR,N  TVSENS.DATAOT
@FOR,N  TVSENS.SUMARY
@FOR,N  TVSENS.AFUNC
@FOR,N  TVSENS.BFUNC
@FOR,N  TVSENS.CFUNC
@FOR,N  TVSENS.DFUNC
@FOR,N  TVSENS.EFUNC
@FOR,N  TVSENS.FNK
@FOR,N  TVSENS.GFUNC
@FOR,N  TVSENS.UFUNC
@FOR,N  TVSENS.VFUNC
@FOR,N  TVSENS.XFUNC
@PACK   TVSENS.
@PREP   TVSENS.
@MAP,N  ,TVSENS.VSEARCH
   IN   TVSENS.
   END
```

Figure 6. Creation of absolute element.

5.2 BATCH PROCESSING

Batch processing of the absolute element VSEARCH is performed whenever the user does not wish interactive communication with the program. If batch processing is desired, the first card image of input data must be the Hollerith string BATCH. This five-character string informs the program that subsequent input will be in an A6, 3X, 3F10.0 format. The alphanumeric information is a mnemonic instruction that directs the program to a specific section of code. The data values, where applicable, represent the initial, the incremental, and the final values of the appropriate input variable. Table 4 lists the mnemonic instructions, the variable defined, and their default values. It should be noted that although the data assignment instructions may occur in any order within the data deck, the STOP instruction must follow any given set of data assignment instructions. The last card image of the data deck must be the mnemonic instruction FINISH. Figure 7 illustrates a sample deck setup for execution of VSEARCH through batch processing.

Table 4. Batch mnemonics.

Mnemonic	Defines	Default Value
NVIRON	=1., coastal; =2., deep	1.
ISTPE	=1., incandescent; =2., thallium; =3. mercury vapor	1.
IRTPE	=1., vidicon; =2., SIT	1.
WAVEHT	wave height	0.
BTMRUF	bottom roughness	0.
TARGLN	target length	0.
TARGHT	target height	0.
SRSEP	source-receiver separation	0.
SPOWR	source power	0.
SDELF	full source angle	0.
RDELF	full receiver angle	0.
OPTRAN	optic transmission	0.
FNUM	f-number	.5
HEIGHT	range (or height)	0.
MAXSW	sets flag to compute maximum width	–
MAXRAN	sets flag to compute maximum range	–
SINGRN	sets flag to compute single range	–
STOP	stop reading input; end-of-record indicator for input	–
FINISH	stop execution; end-of-file indicator for input	–

```
BATCH
NVIRON          1.00000           .00000            .00000
WAVEHT          2.00000           .00000            .00000
BTMRUF           .50000           .00000            .00000
TARGLN         10.00000           .00000            .00000
TARGHT           .50000           .00000            .00000
SRSEP           1.00000           .00000            .00000
ISTPE           2.00000           .00000            .00000
SPOWR         200.00000           .00000            .00000
SDELF          40.00000           .00000            .00000
IRTPE           1.00000           .00000            .00000
RDELF          40.00000           .00000            .00000
OPTRAN           .90000           .00000            .00000
FNUM            1.50000           .00000            .00000
MAXSW            .00000           .00000            .00000
STOP             .00000           .00000            .00000
NVIRON          2.00000           .00000            .00000
RANGE          65.00000         10.00000         105.00000
MAXSW            .00000           .00000            .00000
STOP             .00000           .00000            .00000
SRSEP           1.00000          1.00000           4.00000
RANGE            .00000           .00000            .00000
MAXSW            .00000           .00000            .00000
STOP             .00000           .00000            .00000
SRSEP           5.00000           .00000            .00000
SPOWR          50.00000         50.00000         105.00000
MAXSW            .00000           .00000            .00000
STOP             .00000           .00000            .00000
SDELF          20.00000           .00000            .00000
RDELF          20.00000           .00000            .00000
MAXSW            .00000           .00000            .00000
STOP             .00000           .00000            .00000
FINISH           .00000           .00000            .00000
```

Figure 7. Sample deck setup for batch execution of VSEARCH.

5.3 DEMAND TERMINAL USE

SCOUTS should be executed via a demand terminal whenever the user wishes interactive communication. A demand run also provides intermediate information that can be useful in determining which input variables should be changed.

Execution of the absolute element is performed by responding to the program's questions. Questions concerning environment, source type and receiver type are answered in an I1 format. Other initial input is in a G10.0 format. After the initial input, SCOUTS asks for direction as to parametric or nonparametric, the calculational mode, whether another run is desired, etc. These questions are also answered in an I1 format.

It should be noted that batch processing of SCOUTS can be performed from a demand terminal by responding 'BATCH' to the first question asked. In this case the succeeding input will have to follow exactly the format of Figure 7.

REFERENCES

1. Naval Undersea Center. NUC TP 303, Handbook of Underwater Imaging System Design, by C. J. Funk, S. B. Bryant and P. J. Heckman, Jr. July 1972.

2. Biberman, L. M. and S. Nudelman. Photoelectronic Imaging Devices, Vol. 2, p. 539. Plenum Press, New York, N. Y., 1971.

3. RCA Commercial Engineering. RCA Electro-Optics Handbook, p. 121. Harrison, N. J., 1974.

APPENDIX

This appendix contains a complete FORTRAN IV listing of SCOUTS, arranged in alphabetical order by subroutine name. MAIN is the main program; all the rest are FORTRAN subroutines. All programs have compiled successfully on the NOSC UNIVAC 1110.

```
      FUNCTION AFUNC(A)
C
C
      D3=-1.*ALOG(.00125*(EXP(-A)-EXP(-2.*A))/A)/A
      D1=D3
   10 D2=D3-ALOG(D1)/A
      IF (ABS((D2-D1)/D2).LT..01) GO TO 20
      D1=D2
      GO TO 10
   20 D0=(D2-1.)/2.
      D1=D0/3.
      D4=BFUNC(A,1.)+BFUNC(A,D2)
      D7=BFUNC(A,1.+D0)
      D2=D1*(D4+4.*D7)
      D3=1.
   30 D3=2.*D3
      D1=D1/2.
      D0=D0/2.
      D9=BFUNC(A,1.+D0)
      D8=D9
      E1=EXP(-1.*A*2.*D0)
      J1=D3-1.
      DO 40 J=1,J1
      E2=(1.+(2.*FLOAT(J)-1.)*D0)/(1.+(2.*FLOAT(J)+1.)*D0)
      D8=D8*E1*E2*E2
      D9=D9+D8
   40 CONTINUE
      D5=D1*(D4+2.*D7+4.*D9)
      IF (ABS((D5-D2)/D5).LT..01) GO TO 50
      D2=D5
      D7=D9+D7
      GO TO 30
   50 AFUNC=D5
      RETURN
      END

      FUNCTION BFUNC(A,B)
      BFUNC=(EXP(-A*B))/(B*B)
      RETURN
      END

      FUNCTION CFUNC(IDIR,THET,IWAT)
C
C                COMMON BLOCK OF COMPUTING CONSTANTS
      COMMON /TVCOM/ A(2,2), B(2,11,2), C(20),D(2,2), E(10),
     1 FM(2,2), G(2,2,3), H(2,2,3), N(10), O(15), P(2,2),
     2 Q(15), S(2), T(11), W(10), Z(2), PII
C
      DO 10 II=1,11
      IF (THET.LT.T(II)) GO TO 20
   10 CONTINUE
   20 ID2=MIN0(II,11)
      II=ID2-1
      D5=B(IWAT,II,IDIR)
      CFUNC=D5+((THET-T(II))/(T(ID2)-T(II)))*(B(IWAT,ID2,IDIR)-B(IWAT,II
     1,IDIR))
      RETURN
      END

      SUBROUTINE CONVT (IVAL)
C
C               SUBROUTINE CONVT CONVERTS SINGLE CHARACTER ALPHA-
C          NUMERIC DATA TO INTEGER NUMERICS.
C
      DIMENSION ICHAR(10)
      DATA ICHAR /1H1,1H2,1H3,1H4,1H5,1H6,1H7,1H8,1H9,1H0/
C
      DO 10 I=1,10
      INUM=I
      IF (INUM.EQ.10) INUM=0
      IF (IVAL.EQ.ICHAR(I)) GO TO 20
   10 CONTINUE
      PRINT 40
      GO TO 30
C
   20 IVAL=INUM
   30 RETURN
C
   40 FORMAT (15H NO VALUE FOUND)
      END
```


COPY AVAIL... NOT
PERMIT FULLY ...

```
      SUBROUTINE DATAIN (ISTART,ISUM)
C
C              SUBROUTINE DATAIN READS INPUT DATA
C         AND DEFINES VARIOUS COMPUTING CONSTANTS.
C
C              COMMON BLOCK OF OPTIONS
      COMMON /OPTS/ IRUN,IPARAM,ICAL
C
C              COMMON BLOCK OF I/O VARIABLES
      COMMON /IOLIST/ NVIRON, ISTPE, IRTPE, WAVEHT, BTMRUF, TARGLN,
     1 TARGHT, SRSEP, SPOWR, SDELF,  RDELF, OPTRAN,   FNUM, HEIGHT,
     2 RANGMN, RANGE, RANGAL, RANGER, RANGMX, SWWTHL, SWWTHR, SWWTHR,
     3  SWWTH, AVSWL,  AVSWR,   AVSW, LINCEN, LINEDG, CTRCEN, CTREDG,
     4 NOFXWL, NOFXWR, LIMFLG
C
C              COMMON BLOCK OF COMPUTING CONSTANTS
      COMMON /IVCOM/ A(2,2), B(2,11,2), C(20),D(2,2), E(10),
     1 FM(2,2), G(2,2,3), H(2,2,3), N(10), O(15), P(2,2),
     2 Q(15), S(2), T(11), W(10), Z(2), PTI
C
C              COMMON BLOCK OF UPDATE INFORMATION
      COMMON /UPD/ STEP(14), FDVAL(14)
C
      DIMENSION INST(19)
      REAL LINCEN,LINEDG
C
C              DATA STATEMENT OF INSTRUCTIONS
      DATA INST /6HNVIRON, 6HISTPE , 6HIRTPE , 6HWAVEHT, 6HBTMRUF,
     1             6HTARGLN, 6HTARGHT, 6HSRSEP , 6HSPOWR , 6HSDELF ,
     2             6HRDELF , 6HOPTRAN, 6HFNUM  , 6HRANGE , 6HMAXSW ,
     3             6HMAXRAN, 6HSINGRN, 6HSTOP  , 6HFINISH/
C
C              DEFAULT VALUES FOR INPUT
      DATA NVIRON,ISTPE,IRTPE,WAVEHT,BTMRUF,TARGLN,TARGHT,
     1 SRSEP,SPOWR,SDELF,RDELF,OPTRAN,FNUM /3*1,9*0.,.5/
C
C              DETERMINE THE TYPE OF PROCESSING, BATCH OR DEMAND.  WHEN
C         ISTART=0, THE TYPE OF PROCESSING HAS ALREADY BEEN DETERMINED.
      IF (ISTART.EQ.0) GO TO 20
      IRUN=1
      PRINT 500
C
C              FIRST CARD OF DATA DETERMINES THE FORMAT OF FOLLOWING
C         INPUT - BATCH OR DEMAND TERMINAL.
      READ (5,510) RUN
      IF (RUN.EQ.5HBATCH) GO TO 10
      RUN=6HDEMAND
      IRUN=0
   10 PRINT 520, RUN
C
C              IF BATCH RUN, BRANCH TO INPUT BY MNEMONIC INSTRUCTIONS.
   20 IF (IRUN.EQ.1) GO TO 270
C
C              DEMAND RUN
C
C              IF PREVIOUS CALCULATIONS HAVE BEEN PERFORMED, BRANCH
C         TO CODE THAT DECIDES WHETHER OR NOT NEW CALCULATIONS ARE
C         DESIRED.
      IF (ISTART.EQ.0) GO TO 210
C
C              READ INITIAL VALUES OF INPUT DATA.  IF ISTART=1, PREVIOUS
C         CALCULATIONS HAVE BEEN PERFORMED AND A CHANGE IS BEING MADE ON
C         SPECIFIC INITIAL VALUES.
      PRINT 560
   30 PRINT 570
      READ (5,530) NVIRON
      IF (NVIRON.NE.1.AND.NVIRON.NE.2) GO TO 30
      IF (ISTART.EQ.1) GO TO 240
   40 PRINT 580
      READ (5,540) WAVEHT
      IF (WAVEHT.LT.0.) GO TO 40
      IF (ISTART.EQ.1) GO TO 240
   50 PRINT 590
      READ (5,540) BTMRUF
      IF (BTMRUF.LT.0.) GO TO 50
      IF (ISTART.EQ.1) GO TO 240
      PRINT 600
   60 PRINT 610
      READ (5,540) TARGLN
      IF (TARGLN.LT.0.) GO TO 60
      IF (ISTART.EQ.1) GO TO 240
   70 PRINT 620
      READ (5,540) TARGHT
      IF (TARGHT.LT.0.) GO TO 70
      IF (ISTART.EQ.1) GO TO 240
```


COPY AVAILABLE ... PERMIT FULLY LEGIBLE PRODUCTION

```
      PRINT 630
   80 PRINT 640
      READ (5,540) SRSEP
      IF (SRSEP.LT.0.) GO TO 80
      IF (ISTART.EQ.1) GO TO 240
      PRINT 650
   90 PRINT 660
      READ (5,530) ISTPE
      IF (ISTPE.LT.1.OR.ISTPE.GT.3) GO TO 90
      IF (ISTART.EQ.1) GO TO 240
  100 PRINT 670
      READ (5,540) SPOWR
      IF (SPOWR.LT.0.) GO TO 100
      IF (ISTART.EQ.1) GO TO 240
  110 PRINT 680
      READ (5,540) SDELF
      IF (SDELF.LT.0..OR.SDELF.GT.180.) GO TO 110
      IF (ISTART.EQ.1) GO TO 240
      PRINT 690
  120 PRINT 700
      READ (5,530) IRTPE
      IF (IRTPE.NE.1.AND.IRTPE.NE.2) GO TO 120
      IF (ISTART.EQ.1) GO TO 240
  130 PRINT 710
      READ (5,540) RDELF
      IF (RDELF.LT.0..OR.RDELF.GT.180.) GO TO 130
      IF (ISTART.EQ.1) GO TO 240
  140 PRINT 720
      READ (5,540) OPTRAN
      IF (OPTRAN.LT.0..OR.OPTRAN.GT.1.) GO TO 140
      IF (ISTART.EQ.1) GO TO 240
  150 PRINT 730
      READ (5,540) FNUM
      IF (FNUM.LT..5) GO TO 150
      IF (ISTART.EQ.1) GO TO 240
C
C               PARAMETRIC OR NON-PARAMETRIC MODE?  IF NON-PARAMETRIC
C          MODE (IPARAM .NE. 1), VARIABLE INPUT COMPLETE.
      PRINT 740
      READ (5,530) IPARAM
      IF (IPARAM.NE.1) GO TO 200
C
C               GIVE INDEX OF SPECIFIC PARAMETRIC VARIABLE.  THEN READ
C          ITS INCREMENTAL AND ENDING VALUES.
      PRINT 750
  160 READ (5,530) IN,IN2
      IF (IN.EQ.0) GO TO 180
      IF (IN2.EQ.1H ) GO TO 170
      CALL CONVT (IN2)
      IN=10*IN+IN2
  170 PRINT 760
      READ (5,540) STEP(IN)
      PRINT 770
      READ (5,540) EDVAL(IN)
  180 IF (ISTART.EQ.1) GO TO 260
C
C               IF ADDITIONAL PARAMETRIC VARIABLES ARE DESIRED, CHANGE
C          INDEX.
      PRINT 840
      READ (5,530) IN2
      IF (IN2.NE.1) GO TO 190
      PRINT 830
      GO TO 160
C
C               IF PARAMETRIC MODE, DETERMINE THE CALCULATIONS TO BE
C          PERFORMED.
C               READ THE INITIAL VALUE FOR COMPUTING A SINGLE RANGE, AND
C          THE CORRESPONDING INCREMENTAL AND ENDING VALUES.
  190 IF (IPARAM.NE.1) GO TO 490
      PRINT 850
      READ (5,530) ICAL
      IF (ICAL.GT.4) ICAL=1
      IF (ISTART.EQ.1) GO TO 490
C
C               VARIABLE INPUT COMPLETE.
  200 PRINT 780
      ISTART=1
      GO TO 490
C
C               PREVIOUS CALCULATIONS HAVE BEEN PERFORMED.
C
C               IF ANOTHER RUN IS DESIRED, MAKE NECESSARY CHANGES(IF ANY)
  210 PRINT 790
      READ (5,530) IN2
```


BEST AVAILABLE COPY

```
001          IF (IN2.NE.1) GO TO 490
001          IF (IPARAM.EQ.1) CALL SUMARY (IRUN,ISUM)
001          ISTART=1
001          PRINT 740
001          READ (5,530) IPARAM
001          PRINT 800
001          READ (5,530) IN2
001          IF (IN2.NE.1) GO TO 490
001    C
001    C              GIVE INDEX OF VARIABLE TO BE CHANGED; THEN BRANCH TO
001    C         APPROPRIATE READ.
001          PRINT 750
001      220 READ (5,530) IN,IN2
001          IF (IN2.EQ.1H ) GO TO 230
001          CALL CONVT (IN2)
001          IN=10*IN+IN2
001      230 GO TO (30,90,120,40,50,60,70,80,100,110,130,140,150,240), IN
001    C
001    C              IF PARAMETRIC MODE,  BRANCH TO CODE THAT READS INCREMENTAL
001    C         AND ENDING VALUES.
001      240 IF (IPARAM.NE.1) GO TO 260
001          PRINT 820
001          READ (5,530) IN2
001          IF (IN2.NE.1) GO TO 250
001          GO TO 170
001      250 STEP(IN)=0.
001    C
001    C              IF ADDITIONAL CHANGES ARE DESIRED, CHANGE INDEX.
001      260 PRINT 810
001          READ (5,530) IN2
001          IF (IN2.NE.1) GO TO 190
001          PRINT 830
001          GO TO 220
001    C
001    C
001    C              BATCH RUN.
001    C
001    C              READ INPUT MNEUMONIC.  IF ANY INCREMENTAL VALUE APPEARS,
001    C         SET THE FLAG FOR PARAMETRIC INPUT.
001      270 READ (5,510) IVAR,Z1,Z2,Z3
001          WRITE (6,550) IVAR,Z1,Z2,Z3
001    C
001          IF (Z1.LT.0.) Z1=0.
001          IF (ABS(Z2).EQ.0.) Z3=Z1
001          IF (ABS(Z2).NE.0.) IPARAM=1
001    C
001    C              TEST THE MNEUMONIC AGAINST THE SET OF INSTRUCTIONS THEN
001    C         BRANCH TO APPROPRIATE SECTION OF CODE.
001          DO 280 I=1,19
001          IP=I
001          IF (IVAR.EQ.INST(I)) GO TO 290
001      280 CONTINUE
001          GO TO 270
001    C
001      290 GO TO (300,360,390,310,320,330,340,350,370,380,400,410,420,470,430
001         1,440,460,480,490), IP
001    C
001    C              ENVIRONMENT - COASTAL OR DEEP
001      300 NVIRON=1
001          IF (Z1.EQ.2.) NVIRON=2
001          STEP(1)=Z2
001          ENDVAL(1)=Z3
001          GO TO 270
001    C
001    C              WAVE HEIGHT
001      310 WAVEHT=Z1
001          STEP(4)=Z2
001          ENDVAL(4)=Z3
001          GO TO 270
001    C
001    C       '      BOTTOM ROUGHNESS
001      320 BTMRUF=Z1
001          STEP(5)=Z2
001          ENDVAL(5)=Z3
001          GO TO 270
001    C
001    C              LENGTH OF TARGET
001      330 TARGLN=Z1
001          STEP(6)=Z2
001          ENDVAL(6)=Z3
001          GO TO 270
001    C
001    C              HEIGHT OF TARGET
001      340 TARGHT=Z1
001          STEP(7)=Z2
```


BEST AVAILABLE COPY

```
      EDVAL(7)=Z3
      GO TO 270
C
C                SOURCE RECEIVER SEPARATION
  350 SRSEP=Z1
      STEP(8)=Z2
      EDVAL(8)=Z3
      GO TO 270
C
C                SOURCE TYPE
  360 ISTPE=1
      IF (Z1.EQ.2.) ISTPE=2
      IF (Z1.EQ.3.) ISTPE=3
      STEP(2)=Z2
      EDVAL(2)=Z3
      GO TO 270
C
C                SOURCE POWER
  370 SPOWR=Z1
      STEP(9)=Z2
      EDVAL(9)=Z3
      GO TO 270
C
C                FULL SOURCE ANGLE
  380 SDELF=Z1
      STEP(10)=Z2
      EDVAL(10)=Z3
      IF (SDELF.GT.180.) SDELF=SDELF-180.
      IF (ABS(Z2).EQ.0.) EDVAL(10)=SDELF
      GO TO 270
C
C                RECEIVER TYPE
  390 IRTPE=1
      IF (Z1.EQ.2.) IRTPE=2
      STEP(3)=Z2
      EDVAL(3)=Z3
      GO TO 270
C
C                FULL RECEIVER ANGLE
  400 RDELF=Z1
      STEP(11)=Z2
      EDVAL(11)=Z3
      IF (RDELF.GT.180.) RDELF=RDELF-180.
      IF (ABS(Z2).EQ.0.) EDVAL(11)=RDELF
      GO TO 270
C
C                TRANSMISSION OF OPTICS
  410 OPTRAN=Z1
      STEP(12)=Z2
      EDVAL(12)=Z3
      IF (OPTRAN.GT.1.) OPTRAN=0.
      IF (ABS(Z2).EQ.0.) EDVAL(12)=OPTRAN
      GO TO 270
C
C                F-NUMBER
  420 FNUM=Z1
      STEP(13)=Z2
      EDVAL(13)=Z3
      IF (FNUM.LT..5) FNUM=.5
      IF (ABS(Z2).EQ.0.) EDVAL(13)=FNUM
      GO TO 270
C
C                SET FLAG TO CALCULATE MAXIMUM WIDTH
  430 IF (ICAL.EQ.2) GO TO 450
      ICAL=3
      GO TO 270
C
C                SET FLAG TO CALCULATE MAXIMUM RANGE
  440 IF (ICAL.EQ.3) GO TO 450
      ICAL=2
      GO TO 270
C
C                SET FLAG TO CALCULATE BOTH MAXIMUM WIDTH AND
C          MAXIMUM RANGE
  450 ICAL=4
      GO TO 270
C
C                  SET FLAG TO CALCULATE SINGLE RANGE ONLY
  460 ICAL=1
      GO TO 270
C
C                RANGE
  470 HEIGHT=Z1
      STEP(14)=Z2
      EDVAL(14)=Z3
      GO TO 270
```


BEST AVAILABLE COPY

```
C
C                   SET FLAG TO PERFORM CALCULATIONS; OR
C             RE-WRITE IS IN EFFECT - READ NEW SET OF CHANGES
  480 ISTART=1
C
  490 CONTINUE
      RETURN
C
C
C
  500 FORMAT (46H IF BATCH RUN, ENTER - BATCH; OTHERWISE RETURN)
  510 FORMAT (A6,3X,3F10.0)
  520 FORMAT (1X,A6,4H RUN)
  530 FORMAT (I1,A1)
  540 FORMAT (G10.0)
  550 FORMAT (1X,A6,3X,3F15.5)
  560 FORMAT (27X,22HCONVENTIONAL TV SYSTEM/3X,11HENVIRONMENT)
  570 FORMAT (26H COASTAL OR DEEP, 1 OR 2:?)
  580 FORMAT (32H WAVE HEIGHT-PEAK TO TROUGH,FT:?)
  590 FORMAT (37H BOTTOM ROUGHNESS-PEAK TO TROUGH,FT:?)
  600 FORMAT (3X,17HTARGET DIMENSIONS)
  610 FORMAT (22H LENGTH OF TARGET,FT:?)
  620 FORMAT (22H HEIGHT OF TARGET,FT:?)
  630 FORMAT (3X,24HPLATFORM CHARACTERISTICS)
  640 FORMAT (32H SOURCE RECEIVER SEPARATION,FT:?)
  650 FORMAT (3X,22HSOURCE CHARACTERISTICS)
  660 FORMAT (61H INCANDESCENT, THALLIUM IODIDE, OR MERCURY VAPOR; 1, 2
     1OR 3:?)
  670 FORMAT (21H SOURCE POWER,WATTS:?)
  680 FORMAT (24H FULL SOURCE ANGLE,DEG:?)
  690 FORMAT (3X,24HRECEIVER CHARACTERISTICS)
  700 FORMAT (25H VIDICON OR SIT, 1 OR 2:?)
  710 FORMAT (26H FULL RECEIVER ANGLE,DEG:?)
  720 FORMAT (25H TRANSMISSION OF OPTICS:?)
  730 FORMAT (11H F-NUMBER:?)
  740 FORMAT (44H PARAMETRIC OR NON-PARAMETRIC MODE, 1 OR 2:?)
  750 FORMAT (46H WATER TYPE(1),SOURCE TYPE(2),RECEIVER TYPE(3)/52H WAVE
     1 HEIGHT(4),BOTTOM ROUGHNESS(5),TARGET LENGTH(6)/51H TARGET HEIGHT(
     27),S-R SEPARATION(8),SOURCE POWER(9)/36H SOURCE ANGLE(10),RECEIVER
     3 ANGLE(11)/47H OPTICS TRANSMISSION(12),F-NUMBER(13),RANGE(14)/8H I
     4NDEX:?)
  760 FORMAT (40H INCREMENTAL VALUE (IN FLOATING POINT):?)
  770 FORMAT (35H ENDING VALUE (IN FLOATING POINT):?)
  780 FORMAT (25H PARAMETER INPUT COMPLETE/)
  790 FORMAT (43H DO YOU WANT ANOTHER RUN, YES(1) OR NO(2):?)
  800 FORMAT (39H CHANGE PARAMETER(S), YES(1) OR NO(2):?)
  810 FORMAT (32H MORE CHANGES, YES(1) OR NO(2):?)
  820 FORMAT (48H PARAMETRIC OR NON-PARAMETRIC VARIABLE, 1 OR 2:?)
  830 FORMAT (14H CHANGE INDEX:)
  840 FORMAT (45H MORE PARAMETRIC VARIABLES, YES(1) OR NO(2):?)
  850 FORMAT (54H MAX. RANGE(2), MAX. WIDTH(3), BOTH(4) OR NEITHER(5):?)
      END


      SUBROUTINE DATAOT (ISUM)
C
C                SUBROUTINE DATAOT OUTPUTS THE FOLLOWING VARIABLES:
C         MAXIMUM RANGE, MAXIMUM SWATH WIDTH, CONTRAST AT CENTER,
C         CONTRAST AT EDGE, AVAILABLE LINES AT CENTER, AVAILABLE
C         LINES AT EDGE
C
C                COMMON BLOCK OF OPTIONS
      COMMON /OPTS/ IRUN,IPARAM,ICAL
C
C                COMMON BLOCK OF I/O VARIABLES
      COMMON /IOLIST/ NVIRON, ISTPE, IRTPE, WAVEHT, BTMRUF, TARGLN,
     1 TARGHT, SRSEP, SPOWR, SDELF,  RDELF, OPTRAN,   FNUM, HEIGHT,
     2 RANGMN, RANGE, RANGAL, RANGER, RANGMX, SWWTHL, SWWTHR, SWWTHH,
     3  SWWTH, AVSWL,  AVSWR,   AVSW, LINCEN, LINEDG, CTRCEN, CTREDG,
     4 NDEXWL, NDEXWR, LIMFLG
C
      DIMENSION IA(5)
      DIMENSION ZERO(16)
      REAL LINCEN,LINEDG
C
      DATA IA/6HPOWER ,6HR. ANG,6HS. ANG,6HRKSKTR,6HDISPLA/
C
      EQUIVALENCE (ZERO(1),RANGMN)
C
      IF (LIMFLG.LT.1) GO TO 10
      NDEXWL=5
      NDEXWR=5
C
   10 IF (IRUN.EQ.1.OR.IPARAM.EQ.1) GO TO 70
```


BEST AVAILABLE COPY

```
      NAMF1=6HCOASTA
      NAMF2=6HL
      IF (NVIRON.EQ.1) GO TO 20
      NAMF1=6HDEEP
      NAMF2=6H
   20 NAMF3=6HINLAND
      NAMF4=6H.
      IF (ISTPE-2) 50,30,40
   30 NAMF3=6HTHAL.
      NAMF4=6HIOD.
      GO TO 50
   40 NAMF3=6HMERC.
      NAMF4=6HVAP.
   50 NAMF5=6HVIDICO
      NAMF6=6HN
      IF (IRTPE.EQ.1) GO TO 60
      NAMF5=6HSIT
      NAMF6=6H
C
   60 PRINT 90, NAMF1,NAMF2,NAME3,NAME4
      PRINT 100, WAVEHT,SPOWR
      PRINT 110, BTMRUF,SDELF
      PRINT 120, TARGLN,NAME5,NAME6
      PRINT 130, TARGHT,RDELF
      PRINT 140, OPTRAN
      PRINT 150, SRSEP,FNUM
      PRINT 160, RANGE,RANGAL
      PRINT 170, LINCEN,LINEDG
      PRINT 180, CTRCEN,CTREDG
      PRINT 190, AVSWL,SWWTHL
      PRINT 200, IA(NDEXWL)
      PRINT 210, AVSWR,SWWTHR
      PRINT 200, IA(NDEXWR)
      PRINT 220, AVSW,SWWTH
C
   70 IF (ISUM.EQ.1) REWIND 9
      WRITE (9) NVIRON,WAVEHT,BTMRUF
      WRITE (9) TARGLN,TARGHT
      WRITE (9) SRSEP
      WRITE (9) ISTPE,SPOWR,SDELF
      WRITE (9) IRTPE,RDELF,OPTRAN,FNUM
      WRITE (9) RANGE,RANGAL,RANGMN,RANGMX,RANGEB
      WRITE (9) LINCEN,LINEDG,CTRCEN,CTREDG
      WRITE (9) AVSWL,SWWTHL,IA(NDEXWL)
      WRITE (9) AVSWR,SWWTHR,IA(NDEXWR)
      WRITE (9) AVSW,SWWTH,SWWTHB
C
C
      DO 80 I=1,16
      ZERO(I)=0.
   80 CONTINUE
      NDEXWL=0
      NDEXWR=0
      LIMFLG=0
C
      RETURN
C
C
C
   90 FORMAT (27X,22HCONVENTIONAL TV SYSTEM/13H WATER TYPE: ,2A6,14X,8HS
     1OURCE: ,2A6)
  100 FORMAT (16H WAVE HEIGHT,FT:,F12.5,11X,13HSOURCE POWER:,F12.5)
  110 FORMAT (11H BOTTOM,FT:,F12.5,16X,12HSOURCE BEAM:,F12.5)
  120 FORMAT (11H LENGTH,FT:,F12.5,16X,9HRECEIVER:,2A6)
  130 FORMAT (11H HEIGHT,FT:,F12.5,16X,14HRECEIVER BEAM:,F12.5)
  140 FORMAT (5H TAU:,F12.5)
  150 FORMAT (11H S.R. SEP.:,F12.5,16X,3HF/:,F12.5)
  160 FORMAT (/10H RANGE,FT:,F12.5,17X,9HRANGE,AL:,F12.5)
  170 FORMAT (/25H AVAILABLE LINES AT CTR.:,F12.5,2X,8HAT EDGE:,F12.5)
  180 FORMAT (20H CONTRAST AT CENTER:,F12.5,7X,8HAT EDGE:,F12.5)
  190 FORMAT (/26H AVAILABLE L H S SW WIDTH=,F9.5,13H FT  ACTUAL =,F9.5,
     13H FT)
  200 FORMAT (12H LIMITED BY:,A6)
  210 FORMAT (/26H AVAILABLE R H S SW WIDTH=,F9.5,13H FT  ACTUAL =,F9.5,
     13H FT)
  220 FORMAT (/20H AVAILABLE SW WIDTH=,F9.5,13H FT  ACTUAL =,F9.5,3H FT)
      END
```

BEST AVAILABLE COPY

```
      FUNCTION DFUNC(THET,SRSEP,RAN,EANG,IWAT)
C
C           COMMON BLOCK OF COMPUTING CONSTANTS
      COMMON /IVCOM/ A(2,2), B(2,11,2), C(20),D(2,2), E(10),
     1 FM(2,2), G(2,2,3), H(2,2,3), N(10), O(15), P(2,2),
     2 Q(15), S(2), T(11), W(10), Z(2), PII
C
      D1=(ATAN((SRSEP+RAN*TAN(THET-EANG))/RAN)-EANG)
      D2=((D(IWAT,1)/COS(D1+EANG))+(1/COS(THET-EANG)))
      D3=C(1)*EXP(-D2*A(IWAT,1)*RAN)
      D4=(D3*(COS(EANG+D1)**3.)*(COS(THET)**4))/(RAN*RAN)
      DFUNC=D4
      RETURN
      END

      FUNCTION EFUNC(CURR,J)
C
C           COMMON BLOCK OF COMPUTING CONSTANTS
      COMMON /IVCOM/ A(2,2), B(2,11,2), C(20),D(2,2), E(10),
     1 FM(2,2), G(2,2,3), H(2,2,3), N(10), O(15), P(2,2),
     2 Q(15), S(2), T(11), W(10), Z(2), PII
C
      EFUNC=(ALOG((CURR*Z(J))/FM(J,1))/FM(J,2))
      RETURN
      END

      SUBROUTINE FNK (II,IB)
C
C           COMMON BLOCK OF COMPUTING CONSTANTS
      COMMON /IVCOM/ A(2,2), B(2,11,2), C(20),D(2,2), E(10),
     1 FM(2,2), G(2,2,3), H(2,2,3), N(10), O(15), P(2,2),
     2 Q(15), S(2), T(11), W(10), Z(2), PII
C
      ID2=1
      DO 10 ID1=II,IB
      E(ID1)=ID2
   10 ID2=ID2+1
      IB1=IB-1
      DO 20 ID1=II,IB1
      IF (W(ID1).GT.W(ID1+1)) GO TO 20
      W(ID1+1)=W(ID1)
      E(ID1+1)=E(ID1)
   20 CONTINUE
      RETURN
      END

      FUNCTION GFUNC(A,B,RAN,ANGLER,EANG,SRSEP)
C
C           CALCULATE BMSCTR WIDTH
C
      D1=-((ALOG(B/A))/(RAN*TAN(ANGLER+EANG)-(SRSEP/2.)))
      GFUNC=-((ALOG(.07/A))/D1)
      RETURN
      END

      SUBROUTINE HLIMIT (IFLAG)
C
C           SUBROUTINE HLIMIT COMPUTES THE HEIGHT LIMITS
C
C           COMMON BLOCK OF OPTIONS
      COMMON /OPTS/ IRUN,IPARAM,ICAL
C
C           COMMON BLOCK OF I/O VARIABLES
      COMMON /IOLIST/ NVIRON, ISTPE, IRTPE, WAVEHT, BTMRUF, TARGLN,
     1 TARGHT, SRSEP, SPOWR, SDELF,  RDELF, OPTRAN,   FNUM, HEIGHT,
     2 RANGMN, RANGE, RANGAL, RANGER, RANGMX, SWWTHL, SWWTHR, SWWTHH,
     3  SWWTH, AVSWL,  AVSWR,   AVSW, LINCEN, LINEDG, CTRCEN, CTREDG,
     4 NOFXWL, NOFXWR, LIMFLG
C
C           COMMON BLOCK OF COMPUTING CONSTANTS
      COMMON /IVCOM/ A(2,2), B(2,11,2), C(20),D(2,2), E(10),
     1 FM(2,2), G(2,2,3), H(2,2,3), N(10), O(15), P(2,2),
     2 Q(15), S(2), T(11), W(10), Z(2), PII
C
      REAL LINCEN,LINEDG
C
C           REQUIRED DEPTH OF FIELD, FEET
      DFIELD=.5*(WAVEHT+BTMRUF)+TARGHT
C
C           INITIAL ESTIMATE MINIMUM RANGE, FEET
      R1=(.05/A(NVIRON,1))*3.281
```


BEST AVAILABLE COPY

```
      ANGLER=.0087265*RDELF
      ANGLES=.0087265*SDELF
   10 EX=ATAN(SRSEP/(2.*R1))
      IF (AMAX1(EX+ANGLER,EX+ANGLES).LT.(PII*.49)) GO TO 20
      R1=1.05*R1
      GO TO 10
C
   20 T1=DFIELD+(SRSEP/(TAN(ANGLES)+TAN(ANGLER)))
   30 EX=ATAN(SRSEP/(2.*T1))
      T2=DFIELD+(SRSEP/(TAN(ANGLES+EX)+TAN(ANGLER+EX)))
      IF (T2.GT.0.) GO TO 40
      T2=0.
      GO TO 50
C
   40 IF (ABS(T1-T2).LT..001) GO TO 50
      T1=T2
      GO TO 30
C
C               SET MINIMUM RANGE, FEET
   50 RANGMN=AMAX1(T2,R1)
      RANGMN=AMAX1(RANGMN,SRSEP)
C
C
      T1=SDELF*.5
      T2=RDELF*.5
      D(NVIRON,1)=CFUNC(1,T1,NVIRON)
      D(NVIRON,2)=CFUNC(2,T2,NVIRON)
C
      IF (IRUN.EQ.1) GO TO 70
      IF (IFLAG.NE.0) GO TO 70
      PRINT 130, RANGMN
      IF (IPARAM.NE.1) GO TO 60
      PRINT 160
      READ (5,170) HEIGHT
      GO TO 70
   60 PRINT 140
      READ (5,150) ICAL
   70 CONTINUE
      IF (ICAL.EQ.4) GO TO 90
      IF (ICAL-2) 100,90,80
C
C                    GET MAXIMUM SWATH WIDTH
   80 CONTINUE
      CALL TVFUNC (-1.,3)
      RANGER=Q(1)*3.281
      SWATHR=Q(12)*3.281
      IF (IPARAM.EQ.1) GO TO 110
      GO TO 120
C
C                  GET MAXIMUM RANGE
   90 CONTINUE
      CALL TVFUNC (-1.,2)
      RANGMX=Q(1)*3.281
      IF (ICAL.EQ.4) GO TO 80
      IF (IPARAM.EQ.1) GO TO 110
      GO TO 120
C
  100 IF (IRUN.EQ.1.OR.IPARAM.EQ.1) GO TO 110
      PRINT 160
      READ (5,170) HEIGHT
C
C
C                 CALCULATE SINGLE RANGE
  110 CONTINUE
      IF (HEIGHT.EQ.0.) GO TO 120
      CALL TVFUNC (HEIGHT,1)
  120 CONTINUE
      RANGE=Q(1)*3.281
      RANGAL=Q(1)*A(NVIRON,1)
      AVSWL=RANGE*TAN(ANGLER-Q(13))+0.5*SRSEP
      SWATHL=Q(8)*3.281
      AVSWR=RANGE*TAN(ANGLER+Q(13))-0.5*SRSEP
      SWATHR=Q(10)*3.281
      AVSW=AVSWL+AVSWR
      SWATH=Q(12)*3.281
C
C                 DISPLAY LIMIT IN LIMIT FLAG.
C                 STORE RIGHT AND LEFT-HAND SIDE WIDTH INDICES; AVERAGE
C          LINES AT CENTER AND AT EDGE; CONTRAST AT CENTER AND AT EDGE.
      LIMFLG=Q(3)
      NDEXWL=Q(9)
      NDEXWR=Q(11)
      LINCEN=Q(3)
      LINEDG=Q(4)
      CTRCEN=Q(6)
      CTREDG=Q(7)
```


BEST AVAILABLE COPY

```
C
      RETURN
C
C
C
  130 FORMAT (17H MINIMUM RANGE IS,F9.5,5H FEET)
  140 FORMAT (46H ONE RANGE(1),MAX. RANGE(2) OR MAX. WIDTH(3):?)
  150 FORMAT (I1)
  160 FORMAT (11H RANGE,FT:?)
  170 FORMAT (G10.0)
      END
```

****** MAIN ******

```
C          INPUT FILES:
C                UNIT 5 (CARD READER) -
C                  DATA OF VARIABLE INITIALIZATION AS FOLLOWS:
C                    ENVIRONMENT, TARGET DIMENSIONS, PLATFORM CHARACTER-
C                    ISTICS, SOURCE CHARACTERISTICS, AND RECEIVER
C                    CHARACTERISTICS.
C          OUTPUT FILES:
C                UNIT 6 (PRINTER)
C                  INTERMEDIATE AND SUMMARY INFORMATION
C
C               COMMON BLOCK OF COMPUTING CONSTANTS
      COMMON /IVCOM/ A(2,2), B(2,11,2), C(20),D(2,2), E(10),
     1 FM(2,2), G(2,2,3), H(2,2,3), N(10), O(15), P(2,2),
     2 Q(15), S(2), T(11), W(10), Z(2), PII
C
C               COMMON BLOCK OF OPTIONS
      COMMON /OPTS/ IRUN,IPARAM,ICAL
C
C
C               ATTN. CONST.
      DATA ((A(I,J),J=1,2),I=1,2) /.252, .273, .049, .049/ ,
C
C               EFFECTIVE ALFA ARRAY
      DATA (((B(I,J,K),I=1,2),J=1,11),K=1,2) /  1.,  1., .87, .91, .69,
     1 .78, .62, .72, .49, .63, .38, .55, .33, .51, .29, .49, .22, .44,
     2 .20, .42, .17, .41,  1.,  1., .87, .91, .68, .78, .59, .72, .45,
     3 .63, .34, .55, .28, .51, .25, .49, .18, .44, .15, .42, .14, .41/
C
C               NUMBER OF LINES MATRIX
      DATA ((FM(I,J),J=1,2),I=1,2) /.000127,.00863,9.51E-14,.0133/
C
C               SOURCE RECEIVER MATRIX
      DATA (((G(I,J,K),I=1,2),J=1,2),K=1,3) /  .00361, .00106, .002,
     1 .000953, .0136, .00325, .00747, .00304, .00610, .00287, .00292,
     2 .00346/
C
C               SOURCE RECEIVER MATRIX (BACKSCATTER)
      DATA (((H(I,J,K),I=1,2),J=1,2),K=1,3) /.00435, .00281, .00277,
     1 .00219, .0162, .0134, .00922, .00924, .0090,  .00430, .00642,
     2 .00574/
C
C               LAG MATRIX
      DATA ((P(I,J),J=1,2),I=1,2) /3.70, 105.0, 5.79E-5, 169.0/
C
C               SCATTERING CONSTANT
      DATA S /.238, .030'
C
C               ANGLES
      DATA T /0., .05, .5, 1., 2.5, 5., 7.5, 10., 20., 30., 90./
C
C               RECEIVER FACTOR
      DATA Z /1070000.0,1./
C
C               PI
      DATA PII /3.14159265/
C
C               INITIALIZE THE NUMBER OF SUMMARIES COUNTER! AND SET THE
C          EXECUTION-BRANCH FLAG FOR READING THE FIRST CARD OF INPUT.
C          THIS CARD SPECIFIES THE TYPE OF PROCESSING, BATCH OR DEMAND
C          TERMINAL.
      ISTART=-1
      ISUM=0
C
C               READ DATA INPUT.  IF NO DATA IS GIVEN, STOP EXECUTION.
   10 CALL DATAIN (ISTART,ISUM)
      IFLAG=0
```


BEST AVAILABLE COPY

```
      IF (ISTART.NE.1) GO TO 30
C
C             INCREMENT THE NUMBER OF SUMMARIES COUNTER.  PERFORM
C        CALCULATIONS AND OUTPUT RESULTS.
   20 ISUM=ISUM+1
      CALL HLIMIT (IFLAG)
      CALL DATAOT (ISUM)
C
C             RE-SET THE EXECUTION-BRANCH FLAG TO STOP EXECUTION.  IF
C        IN PARAMETRIC MODE, UPDATE THE NECESSARY PARAMETER.  (IF THE
C        VALUE OF A PARAMETER HAS BEEN CHANGED, SUBROUTINE UPDATE WILL
C        SET THE EXECUTION-BRANCH FLAG TO PERFORM NEW CALCULATIONS.)
C        IF IN NON-PARAMETRIC MODE, OR IF NO CHANGE IN THE PARAMETERS
C        IS MADE, CONTINUE READING INPUT.
      ISTART=0
      IF (IPARAM.EQ.1) CALL UPDATE (ISTART)
      IFLAG=ISTART
      IF (ISTART.EQ.1) GO TO 20
      GO TO 10
C
C             IF SUMMARY INFORMATION IS GIVEN, OUTPUT THE SUMMARY
C        TABLE.  STOP EXECUTION.
   30 IF (ISUM.NE.0) CALL SUMARY (IRUN,ISUM)
      STOP
C
      END


      SUBROUTINE SUMARY (IRUN,ISUM)
C
C             SUBROUTINE SUMARY OUTPUTS COMPLETE SUMMARY INFORMATION
C        ON THE VISUAL SEARCH SYSTEM PROGRAM
C
      DIMENSION  NVIRON(8),WAVEHT(8),RTMRUF(8),TARGLN(8),TARGHT(8),
     1  SRSEP(8), ISTPE(8), SPOWR(8), SDELF(8), IRTPE(8),
     2  RDELF(8),OPTRAN(8),  FNUM(8), AVSWR(8),  AVSW(8), RANGE(8),
     3 RANGMN(8),RANGMX(8),RANGEB(8), SWWTH(8),SWWTHL(8),SWWTHR(8),
     4 SWWTHR(8),RANGAL(8),LINCEN(8),LINEDG(8),CTRCEN(8),CTREDG(8),
     5  AVSWL(8),    IAL(8),    IAR(8),IPART2(8)
      DIMENSION NAME(14)
      REAL LINCEN,LINEDG
C
      DATA NAME /6H   COA,4HSTAL,6H       ,4HOFEP,
     1            6H   INC,4HAND.,6HTHAL. ,4HIOD.,
     2            6HMERC. ,4HVAP.,6H   VID,4HICON,
     3            6H       ,4H SIT/
C
      REWIND 9
C
      ICK=4
      IF (IRUN.EQ.1) ICK=8
      KNTR=0
      DO 70 I=1,ISUM
      KNTR=KNTR+1
      READ (9) NVIRON(KNTR),WAVEHT(KNTR),RTMRUF(KNTR)
      READ (9) TARGLN(KNTR),TARGHT(KNTR)
      READ (9) SRSEP(KNTR)
      READ (9) ISTPE(KNTR),SPOWR(KNTR),SDELF(KNTR)
      READ (9) IRTPE(KNTR),RDELF(KNTR),OPTRAN(KNTR),FNUM(KNTR)
      READ (9) RANGE(KNTR),RANGAL(KNTR),RANGMN(KNTR),RANGMX(KNTR),RANGEB
     1(KNTR)
      READ (9) LINCEN(KNTR),LINEDG(KNTR),CTRCEN(KNTR),CTREDG(KNTR)
      READ (9) AVSWL(KNTR),SWWTHL(KNTR),IAL(KNTR)
      READ (9) AVSWR(KNTR),SWWTHR(KNTR),IAR(KNTR)
      READ (9) AVSW(KNTR),SWWTH(KNTR),SWWTHR(KNTR)
C
      IF (KNTR.EQ.ICK.OR.I.EQ.ISUM) GO TO 10
      GO TO 70
C
   10 CONTINUE
      IF (IRUN.EQ.1) GO TO 20
      PRINT 80
      READ (5,90)
      GO TO 30
   20 PRINT 100
   30 CONTINUE
C
      DO 40 K=1,KNTR
      IN=NVIRON(K)+NVIRON(K)-1
      INP1=IN+1
      NVIRON(K)=NAME(IN)
      IPART2(K)=NAME(INP1)
   40 CONTINUE
```


BEST AVAILABLE COPY

```
      PRINT 110, ((NVIRON(K),IPART2(K)),K=1,KNTR)
      PRINT 120, (WAVEHT(K),K=1,KNTR)
      PRINT 130, (BTMRUF(K),K=1,KNTR)
      PRINT 140, (TARGLN(K),K=1,KNTR)
      PRINT 150, (TARGHT(K),K=1,KNTR)
      PRINT 160, (SRSEP(K),K=1,KNTR)
      DO 50 K=1,KNTR
      IN=ISTPE(K)+ISTPE(K)+3
      INP1=IN+1
      ISTPE(K)=NAME(IN)
      IPART2(K)=NAME(INP1)
   50 CONTINUE
      PRINT 170, ((ISTPE(K),IPART2(K)),K=1,KNTR)
      PRINT 180, (SPOWR(K),K=1,KNTR)
      PRINT 190, (SDELF(K),K=1,KNTR)
      DO 60 K=1,KNTR
      IN=IRTPE(K)+IRTPE(K)+9
      INP1=IN+1
      IRTPE(K)=NAME(IN)
      IPART2(K)=NAME(INP1)
   60 CONTINUE
      PRINT 200, ((IRTPE(K),IPART2(K)),K=1,KNTR)
      PRINT 210, (RDELF(K),K=1,KNTR)
      PRINT 220, (OPTRAN(K),K=1,KNTR)
      PRINT 230, (FNUM(K),K=1,KNTR)
      PRINT 240, (RANGE(K),K=1,KNTR)
      PRINT 250, (RANGAL(K),K=1,KNTR)
      PRINT 260, (RANGMN(K),K=1,KNTR)
      PRINT 270, (RANGMX(K),K=1,KNTR)
      PRINT 280, (RANGEB(K),K=1,KNTR)
      PRINT 290, (LINCEN(K),K=1,KNTR)
      PRINT 300, (LINFDG(K),K=1,KNTR)
      PRINT 310, (CTRCEN(K),K=1,KNTR)
      PRINT 300, (CTRFDG(K),K=1,KNTR)
      PRINT 320, (AVSWL(K),K=1,KNTR)
      PRINT 330, (SWWTHL(K),K=1,KNTR)
      PRINT 340, (IAL(K),K=1,KNTR)
      PRINT 350, (AVSWR(K),K=1,KNTR)
      PRINT 330, (SWWTHR(K),K=1,KNTR)
      PRINT 340, (IAR(K),K=1,KNTR)
      PRINT 360, (AVSW(K),K=1,KNTR)
      PRINT 330, (SWWTH(K),K=1,KNTR)
      PRINT 280, (SWWTHB(K),K=1,KNTR)
C
      KNTR=0
   70 CONTINUE
      ISUM=0
C
      RETURN
C
C
C
   80 FORMAT (24H CLEAR SCREEN AND RETURN)
   90 FORMAT (80A1)
  100 FORMAT (1H1)
  110 FORMAT (12H WATER TYPE:,11X,8(2X,A6,A4))
  120 FORMAT (16H WAVE HEIGHT,FT:,7X,8F12.4)
  130 FORMAT (21H BOTTOM ROUGHNESS,FT:,2X,8F12.4)
  140 FORMAT (18H TARGET LENGTH,FT:,5X,8F12.4)
  150 FORMAT (18H TARGET HEIGHT,FT:,5X,8F12.4)
  160 FORMAT (13H S.R.SEP.,FT:,10X,8F12.4)
  170 FORMAT (8H SOURCE:,15X,8(2X,A6,A4))
  180 FORMAT (20H SOURCE POWER,WATTS:,3X,8F12.4)
  190 FORMAT (17H SOURCE BEAM,DEG:,6X,8F12.4)
  200 FORMAT (10H RECEIVER:,13X,8(2X,A6,A4))
  210 FORMAT (19H RECEIVER BEAM,DEG:,4X,8F12.4)
  220 FORMAT (5H TAU:,18X,8F12.4)
  230 FORMAT (10H F-NUMBER:,13X,8F12.4/)
  240 FORMAT (10H RANGE,FT:,13X,8F12.4)
  250 FORMAT (2X,10H RANGE,AL:,11X,8F12.4)
  260 FORMAT (2X,12H MINIMUM,FT:,9X,8F12.4)
  270 FORMAT (2X,12H MAXIMUM,FT:,9X,8F12.4)
  280 FORMAT (2X,9H BEST,FT:,12X,8F12.4)
  290 FORMAT (17H AV LINES AT CTR:,6X,8F12.4)
  300 FORMAT (2X,9H AT EDGE:,12X,8F12.4)
  310 FORMAT (17H CONTRAST AT CTR:,6X,8F12.4)
  320 FORMAT (23H AV L.H.S. SW WIDTH,FT:,8F12.4)
  330 FORMAT (2X,11H ACTUAL,FT:,10X,8F12.4)
  340 FORMAT (2X,14H - LIMITED BY:,7X,8(6X,A6))
  350 FORMAT (23H AV R.H.S. SW WIDTH,FT:,8F12.4)
  360 FORMAT (19H AV SWATH WIDTH,FT:,4X,8F12.4)
      END
```


BEST AVAILABLE COPY

```
      SUBROUTINE TVFUNC (RSET,ICAL)
C
C
C               COMMON BLOCK OF I/O VARIABLES
      COMMON /IOLIST/ NVIRON, ISIPE, IRTPE, WAVEHT, BTMRUF, TARGLN,
     1 TARGHT, SRSEP, SPOWR, SDELF,  RDELF, OPTRAN,   FNUM, HEIGHT,
     2 RANGMN, RANGE, RANGAL, RANGER, RANGMX, SWWTHL, SWWTHR, SWWTHR,
     3  SWWTH, AVSWL,  AVSWR,   AVSW, LINCEN, LINEDG, CTRCEN, CTREDG,
     4 NOEXWL, NOEXWR, LIMFLG
C
C               COMMON BLOCK OF COMPUTING CONSTANTS
      COMMON /IVCOM/ A(2,2), B(2,11,2), C(20),D(2,2), E(10),
     1 FM(2,2), G(2,2,3), H(2,2,3), N(10), O(15), P(2,2),
     2 Q(15), S(2), T(11), W(10), Z(2), PII
C
      REAL LINCEN,LINEDG
C
      I=NVIRON
      J=IRTPE
      K=ISTPE
C
C               COMPUTE IN METERS:  DEPTH OF FIELD, SOURCE-RECEIVER
C           SEPARATION, AND VELOCITY(METERS/SEC).
      DF=.3048*(.5*(WAVEHT+BTMRUF)+TARGHT)
      ANGLER=.00872665*RDELF
      ANGLES=.00872665*SDELF
      SRSEP1=.3048*SRSEP
      SRSEP2=.5*SRSEP1
C
C               COMPUTE SIGNAL CONSTANT AND BACKSCATTER CONSTANT
      C(1)=.00083025*SPOWR*OPTRAN*G(I,J,K)
      C(1)=C(1)/(128.*PII*(1.0-COS(ANGLES))*FNUM*FNUM)
      C(2)=C(1)*S(I)*.02*H(I,J,K)/(1.50*G(I,J,K))
C
C               INITIALIZE N, Q, AND O
      DO 10 IT=1,10
      N(IT)=0
      Q(IT)=0.
      O(IT)=0.
   10 CONTINUE
      DO 20 IT=11,15
      Q(IT)=0.
      O(IT)=0.
   20 CONTINUE
C
      Q(2)=-ANGLER
      O(2)=Q(2)
      IFLAG=0
      IFLAG2=0
      N(1)=ICAL
C
C               SET OR CHOOSE RANGE IN METERS
      RANG=.3048*RSET
      IF (ICAL.EQ.1) GO TO 30
      RANG1=.3346*RANGMN
      RANG=RANG1
   30 C(4)=1./A(1,1)
   40 O(1)=RANG
      EANG=ATAN(SRSEP1/(RANG+RANG))
      DELTA1=TAN(ANGLER-EANG)
      DELTA2=TAN(ANGLER+EANG)
      C(5)=6.*RANG*(DELTA1+DELTA2)
      C(5)=C(5)/(.3048*TARGLN)
      IF (C(5).LT.525.) GO TO 50
      N(3)=1
      O(12)=0.
      GO TO 150
   50 N(4)=1
      N(3)=0
      T1=0.
      T2=0.
C               COMPUTE:  SIGNAL CURRENT AT CENTER, OBJECT LINES AT
C           CENTER, AND OBJECT LINES AT EDGE
      C(6)=DFUNC(0.,SRSEP1,RANG,EANG,NVIRON)
      T5=FFUNC(C(6),IRTPE)
      O(3)=AMIN1((T5*8.)/C(5),4200./C(5))
      T5=-ANGLER
      C(7)=DFUNC(T5,SRSEP1,RANG,EANG,NVIRON)
      T5=FFUNC(C(7),IRTPE)
      O(4)=AMIN1((T5*8.)/C(5),4200./C(5))
C
      IF (T5.GT.C(5)) GO TO 110
      T1=1.
C
C               COMPUTE SIGNAL CURRENT AND NUMBER OF LINES
```


BEST AVAILABLE COPY

```
      T5=-ANGLER
   60 T6=DFUNC(T5,SRSEP1,RANG,EANG,NVIRON)
      T6=FFUNC(T6,IRIPE)
      IF (T6.GT.C(5)) GO TO 70
      T3=1.
      GO TO 80
C
C                 SIGNAL LIMITED?
   70 T3=0.
   80 IF (T3.LT..5) GO TO 100
      T1=T3
      T5=T5+.05*(ANGLER+EANG)
      IF (ABS(T5-EANG).LT.(.5*EANG)) GO TO 90
      GO TO 60
C
C                 ZERO SWATH WIDTH DUE TO POWER LIMIT
   90 O(5)=1.
      W(1)=SRSEP2
      W(4)=-SRSEP2
      O(2)=0.
      GO TO 120
C
C                 NORMAL EXIT FROM POWER LIMIT
  100 O(5)=1.
      O(2)=T5-(.05*(ANGLER+EANG))
      T4=RANG*TAN(EANG-T5)
      W(1)=T4+SRSEP2
      W(4)=T4-SRSEP2
      GO TO 120
C
C                 MAXIMUM WIDTH
  110 W(1)=RANG*DELTA1+SRSEP2
      W(4)=RANG*DELTA2-SRSEP2
      O(5)=1.
C
  120 T1=XFUNC(0.,SRSEP1,EANG,RANG,ANGLES,NVIRON)
      O(6)=.5*C(6)/(C(6)+T1)
      T1=XFUNC(-ANGLER,SRSEP1,EANG,RANG,ANGLES,NVIRON)
      O(7)=.5*C(7)/(C(7)+T1)
      IF (O(7).LT..07) GO TO 130
      W(7)=RANG*DELTA2-SRSEP2
      GO TO 140
C
  130 W(7)=GFUNC(O(6),O(7),RANG,ANGLER,EANG,SRSEP1)
  140 W(2)=VFUNC(DELTA1,RANG,DF,SRSEP1)
      T4=TAN(ANGLES+EANG)
      W(3)=UFUNC(T4,RANG,DF,SRSEP1)
      W(5)=UFUNC(DELTA2,RANG,DF,SRSEP1)
      T4=TAN(ANGLES-EANG)
      W(6)=VFUNC(T4,RANG,DF,SRSEP1)
      CALL FNK (1,3)
      O(8)=W(3)
      O(9)=F(3)
      CALL FNK (4,7)
      O(10)=W(7)
      O(11)=F(7)
      O(12)=O(8)+O(10)
      IF (O(12).LT.0.) O(12)=0.
      O(13)=EANG
C
  150 CONTINUE
      IF (IFLAG.EQ.1) GO TO 160
      IF (ICAL.EQ.1) GO TO 230
      IF (O(12).LE.0.) GO TO 230
      IFLAG=1
      GO TO 230
C
  160 IF (ABS(C(4)).GE..05/A(I,1)) GO TO 170
      IFLAG=0
      GO TO 230
C
  170 IF (ICAL-2) 230,180,200
  180 IF (O(12).GT.0.) GO TO 190
      IFLAG2=1
      C(4)=-.5*ABS(C(4))
      GO TO 230
C
  190 IF (IFLAG2.EQ.0) GO TO 230
      C(4)=.5*ABS(C(4))
      GO TO 230
C
  200 IF (O(12).GT.Q(12)) GO TO 230
      IF (O(12).GT.0.) GO TO 220
      IF (RANG-1.5*ABS(C(4)).GT.RANG1) GO TO 210
```


BEST AVAILABLE COPY

```
      RANG=RANG-ABS(C(4))
      C(4)=.5*ABS(C(4))
      GO TO 230
C
  210 RANG=RANG-2.*ABS(C(4))
      C(4)=.5*ABS(C(4))
      GO TO 230
C
  220 C(4)=-.5*C(4)
  230 DO 240 II=1,13
      W(II)=U(II)
  240 CONTINUE
      IF (IFLAG.EQ.0) GO TO 250
      RANG=RANG+C(4)
      GO TO 40
C
  250 RETURN
      END
```

```
      FUNCTION UFUNC(A,RAN,DFIELD,SRSEP)
      UFUNC=A*(RAN-(SRSEP/(2.*A))-DFIELD)
      RETURN
      END
```

```
      SUBROUTINE UPDATE (ISTART)
C
C
C                    COMMON BLOCK OF I/O VARIABLES
      COMMON /IOLIST/ NVIRON, ISTPE, IRTPE, WAVEHT, UTMRUF, TARGLN,
     1 TARGHT, SRSEP, SPOWR, SUELF,  RUELF, OPTRAN,   FNUM, HEIGHT,
     2 RANGMN, RANGE, RANGAL, RANGER, RANGMX, SWWTHL, SWWTHR, SWWTHR,
     3  SWWTH, AVSWL,  AVSWR,    AVSW, LINCEN, LINEDG, CTRCEN, CTREDG,
     4 NUEXWL, NUEXWR, LIMFLG
C
C                    COMMON BLOCK OF UPDATE INFORMATION
      COMMON /UPD/ STEP(14), EDVAL(14)
C
      DIMENSION VALUE(14)
      DIMENSION BEG(14),ID(14)
      DATA NTRY /0/
C
      EQUIVALENCE (VALUE(4),WAVEHT)
C
      VALUE(1)=NVIRON
      VALUE(2)=ISTPE
      VALUE(3)=IRTPE
      IF (NTRY.NE.0) GO TO 20
      NUM=0
      DO 10 I=1,14
      IF (ABS(STEP(I)).EQ.0.) GO TO 10
      NUM=NUM+1
      ID(NUM)=I
      BEG(NUM)=VALUE(I)
   10 CONTINUE
      NTRY=1
C
   20 IF (NUM.EQ.0) GO TO 70
      I=1
   30 IN=ID(I)
      IF (VALUE(IN).NE.EDVAL(IN)) GO TO 50
      IF (I.NE.NUM) GO TO 40
      NTRY=0
      GO TO 70
   40 VALUE(IN)=BEG(I)
      I=I+1
      GO TO 30
C
   50 ISTART=1
      VALUE(IN)=VALUE(IN)+STEP(IN)
      IF (ABS(STEP(IN)).EQ.STEP(IN)) GO TO 60
      IF (VALUE(IN).LT.EDVAL(IN)) VALUE(IN)=EDVAL(IN)
      GO TO 70
   60 IF (VALUE(IN).GT.EDVAL(IN)) VALUE(IN)=EDVAL(IN)
C
   70 NVIRON=VALUE(1)
      ISTPE=VALUE(2)
      IRTPE=VALUE(3)
C
      RETURN
C
      END
```


BEST AVAILABLE COPY

```
      FUNCTION VFUNC(A,RAN,DFIELD,SRSEP)
C              CALCULATE SOURCE R. AND RECEIVER L WIDTH
      VFUNC=A*(RAN+(SRSEP/(2.*A))-DFIELD)
      RETURN
      END
```

```
      FUNCTION XFUNC(THET,SRSEP,EANG,RAN,ANG,I)
C
C              COMMON BLOCK OF COMPUTING CONSTANTS
      COMMON /IVCOM/ A(2,2), B(2,11,2), C(20),D(2,2), E(10),
     1 FM(2,2), G(2,2,3), H(2,2,3), N(10), O(15), P(2,2),
     2 Q(15), S(2), T(11), W(10), Z(2), PII
C
      F1=(SRSEP/(TAN(EANG+ANG)+TAN(EANG-THET)))
      F2=COS(THET-EANG)
      F3=((D(I,1)*A(1,1))+(D(I,2)*A(1,2)))/F2
      F4=((AFUNC(F3*F1))/F1-(AFUNC(F3*RAN)/RAN))
      XFUNC=C(2)*(COS(THET)**4)*F2*F4
      RETURN
      END
```

BEST AVAILABLE COPY